Trabalho 3 – Organização e Arquitetura de Computadores

Aluno(a): Júlia Yuri Garcia Baba
Matrícula: 19/0057921
Turma C

1. Objetivos

O objetivo principal desse trabalho é promover a prática da programação em assembler do RISC-V, que consiste no desenho de linhas na janela gráfica do RARS, através da implementação do algoritmo de Bresenham.

2. Documentação do código

Sendo assim, para o funcionamento do algoritmo deve-se seguir três estratégias básicas: implementação da interface com o usuário, assim como, das funções, por fim, a configuração do bitmap display do RARS (já descrita no relatório).

2.1.Interface com o usuário

Primeiramente imprimiu-se uma mensagem para a solicitação dos valores correspondentes a x0, y0, x1 e y1, tal ação foi feita por meio de chamadas do sistema, da mesma forma que a solicitação das coordenadas.

2.1.Implementação das funções

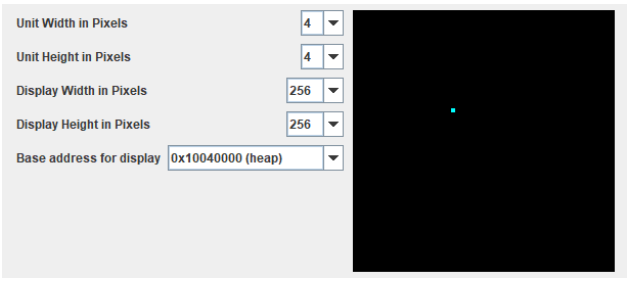
Foram implementadas três funções ao longo do algoritmo:

- *getaddress:* Retorna o endereço de memória correspondente ao pixel, tendo como parêmetros x e y, foi calculada pela equação aritmética:

  $$end = 4*x + 4*(y*dx) + org \quad (1)$$

  Sendo assim, armazena na área de dados a largura, altura e a origem(endereço inicial na memória) da imagem em pixels.

- *ponto:* Desenha um ponto na tela gráfica a partir de suas coordenadas x e y, na cor definida na área de dado, de acordo com o endereço de memória correspondente ao pixel, dado pela função anterior.



1. Funcionamento da função ponto – coordenadas(24,24)

- *linha:* Desenha a linha de um ponto ao outro, tendo como parâmetros de entrada (X0, Y0) e (X1, Y1), seguindo os passos:
  1. Calcula-se a variaçaõ de dx e dy, dados por:

     $$dx = x1-x0 \quad (2)$$

     $$dy = y1-y0 \quad (3)$$
  2. Calcula-se o parâmetro de decisão, dado por:

     $$D = 2dy – dx \quad (4)$$
  3. Coloca-se nas variáveis de trabalho (x,y) o ponto inicial, nesse caso, a variável de trabalho x também servirá como contador.
  4. Plota-se o ponto, utilizando a função *ponto.*
  5. Nesse ponto do código, faz-se um loop e nele é analisado o valor de D, caso o parâmetro de decisão seja maior ou igual a zero, duas operações são realizadas, além da plotagem do ponto:

     $$y = y + 1 \quad (5)$$

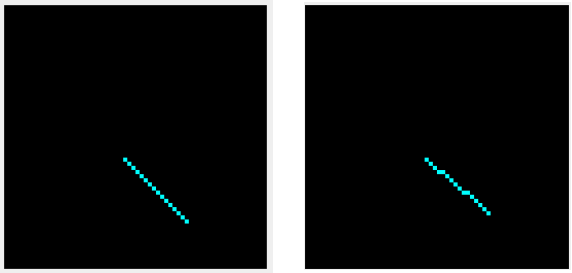
     $$D = D + (2*dy-2*dx) \quad (6)$$

     Caso, seja menor que zero, plota-se o ponto e a operação é feita:

     $$D = D + (2*dy) \quad (7)$$
  6. Repete-se o passo cinco até que o ponto (x1,y1) seja alcançado.

De modo geral, primeiro o programa chama a função linha, os parâmetros são calculados e o primeiro ponto é plotado. Feito isso, os pontos são plotados sucessivamente até que o último ponto seja atingido.

Contudo, durante a execução do programa, percebeu-se uma falha, que pode passar despercebida, dependendo das coordenadas colocadas. Como falado anteriormente, caso D seja maior ou igual a zero faz-se determinado cálculo, no entanto, existe uma parte dessa equação (2*dy-2*dx) que deve ser usada em módulo. Para corrigir isso, foi feita uma outra condição, que faz um subtração ao invés de uma adição, caso dy seja menor que dx.



2. Antes e depois da correção – coordenadas (29,37) e (44,56)