



Arquitetura de Computadores

LIC. EM ENG.^a INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA



Lab 2 – Programação em *Assembly*: Ciclos

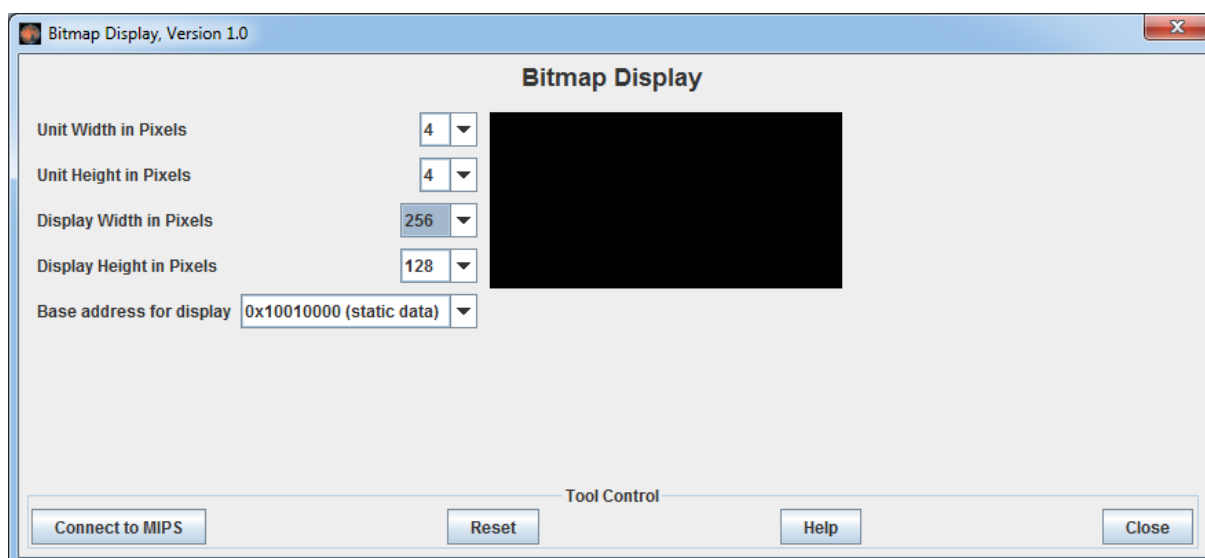
Os exercícios que se seguem devem ser executados através do MARS, um simulador da arquitetura MIPS disponível em <http://courses.missouristate.edu/KenVollmar/MARS/>. Este simulador inclui alguns periféricos simulados acessíveis através do menu **Tools**. Para este trabalho iremos explorar o display gráfico que o MARS simula.

1. Bitmap Display

No menu **Tools** encontra-se disponível uma ferramenta que simula um display gráfico onde a cor de cada pixel corresponde à combinação de valores de vermelho, verde e azul armazenados numa palavra de 32 bits. A cor é definida pelos valores de vermelho (bits 16-23), verde (bits 8-15) e azul (bits 0-7). Os bits 24 a 31 não são utilizados.

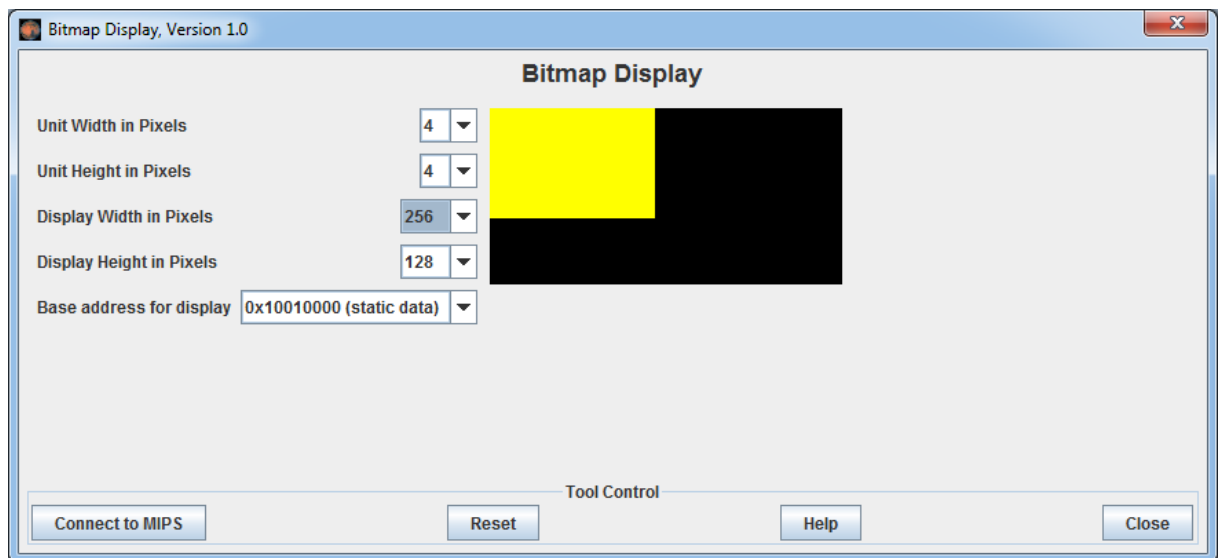
A título exemplificativo, para atribuir a cor amarela a um pixel deve-se escrever na zona de memória correspondente o valor 0x00FFFF00.

Para melhor visualização, vamos escolher um display de 256 por 128 *pixels*. Pelo mesmo motivo, vamos definir que cada um desses *pixels* corresponde a um bloco de *pixels* 4x4 do nosso ecrã. Ou seja, a resolução efetiva do display será de 64 por 32 *pixels*. Configure a ferramenta de acordo com a figura seguinte:

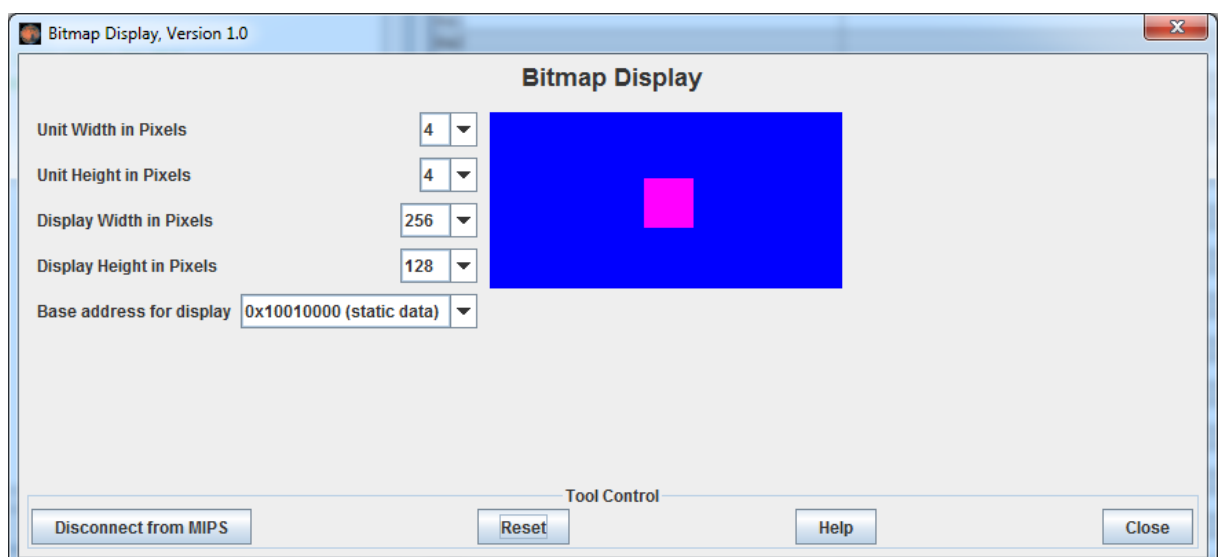


Assim, o primeiro pixel (canto superior esquerdo) está mapeado no endereço base do display, o segundo pixel da primeira linha encontra-se 4 bytes à frente e o primeiro pixel da segunda linha está 64 x 4 bytes adiantado em relação ao endereço base.

- a) Faça um pequeno programa que desenhe um rectângulo amarelo no canto superior esquerdo do ecrã (do simulador) de dimensões 30x20 *pixels*. O resultado obtido deverá ser semelhante ao da figura seguinte:



- b) Escreva agora um programa que desenhe um quadrado magenta (0x00FF00FF) sobre um fundo azul. O quadrado deverá ter dimensões 8x8 e deverá aparecer centrado no ecrã. O resultado deverá ser o seguinte:



- c) Escreva um programa que atribua a cada um dos 64x32 *pixels* uma cor calculada da seguinte forma: vermelho=número da linha multiplicado por 4, verde=número da coluna multiplicado por 4 e azul=0. O resultado deverá ser o seguinte:

