



UNIVERSIDADE TECNOLÓGICA FEDERAL DO
PARANÁ - UTFPR

ELF74 - SISTEMAS EMBARCADOS - S11

Lab 3 - Histograma

Aluno: Mateus Vieira Freitas

PROF.: DOUGLAS PAULO BERTRAND RENAUX

27 de outubro de 2021

1 Introdução

O Laboratório 3 consiste em uma prática cujo objetivo final é desenvolver uma rotina capaz de gerar um histograma de uma imagem em tons de cinza. Um histograma nada mais é que uma representação gráfica da distribuição de tons de uma imagem – lembrando que os pixels em escala de cinza podem variar de 0 (preto) a 255 (branco).

Com o problema bem definido, foi possível partir para o estudo da plataforma que, nesse caso, se trata apenas do *software* – o qual foi desenvolvido em linguagem Assembly, conforme especificado nos requisitos. Esse estudo teve por finalidade obter um maior conhecimento sobre as funções em Assembly para um processador Cortex M4.

2 Estudo da Plataforma

O estudo da plataforma foi feito visando adquirir informações sobre o uso de funções específicas de Assembly, como parametrizar funções, como acessar posições na memória e seus conteúdos, além de lógica básica de programação nessa linguagem (blocos de condicionais, *loops* e definição de funções).

Finalizados os estudos, foi possível partir para o projeto da solução.

3 Projeto da Solução

Após adquirir uma maior familiaridade com o *software* do projeto, o design da solução se tornou mais direcionado. A solução foi pensada de acordo com o seguinte diagrama:

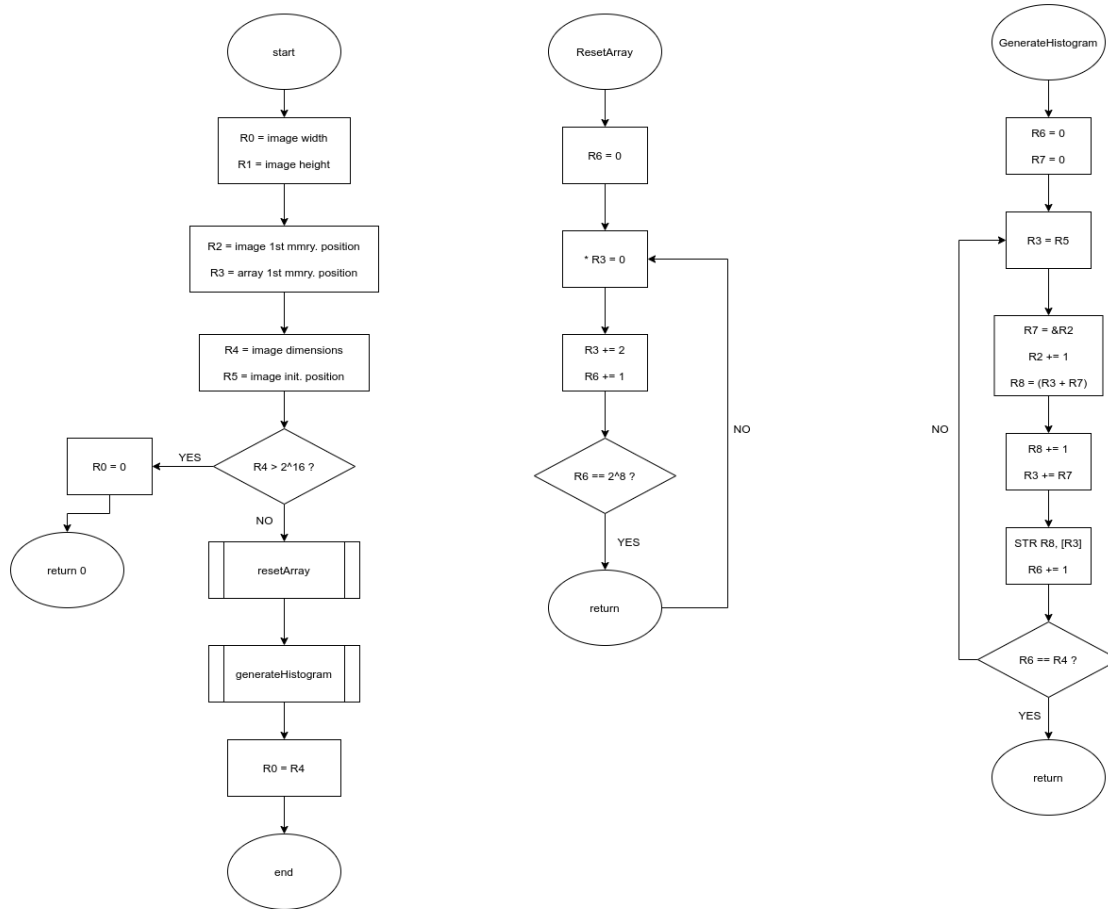


Figura 1: Projeto da solução

Vale-se destacar alguns pontos sobre diagrama acima:

- A condicional $R4 > 2^{16}$ é uma validação necessária devido ao requisito do projeto de o tamanho máximo permitido da imagem ser de 64K (65536) pixels.
- A função de *ResetArray* basicamente percorre o vetor de 256 posições e zera cada um dos elementos.
- A função de *GenerateHistogram* salva a posição inicial do vetor em R3 e salva o valor da posição de memória de R2 (primeira posição da memória da imagem) em R7 e vai percorrendo R2. Além disso há o fluxo responsável por salvar em R8 o valor atual da contagem do valor do pixel na iteração. Isso só se encerra após R6 ser do tamanho da imagem.

4 Codificação

Este laboratório foi realizado em linguagem de programação Assembly, juntamente do C++, sendo esse segundo apenas para chamada de função do assembly, inicialização de alguns parâmetros, e definição da matrix das imagens usadas como caso de teste. Já no assembly, foi feita toda a lógica do projeto, envolvendo branches, condicionais e laços, a fim de conseguir percorrer a matrix da imagem fornecida pelo C++ e gerar o histograma para a mesma.

A codificação foi feita baseando-se no *flowchart* da Figura 1, que não foi tão problemática, pois a modelagem inicial permite um olhar mais assertivo nas abordagens. Após depuração e testes com os casos de uso fornecidos pelo professor, foi possível concluir que o laboratório foi concluído com sucesso, pois os resultados estavam de acordo com os *screenshots* presentes no roteiro. Sendo assim, é possível passar qualquer matrix de imagem em escala de cinza (de até 64K pixels) para o assembly, por meio de definição no arquivo C++, que o projeto irá imprimir no terminal um histograma da imagem fornecida, bem como o tamanho (altura vezes largura) da imagem, em pixels.