

Relatório do Lab 3

Disciplina de Sistemas Embarcados – Prof. Douglas Renaux

Autores: Luis Camilo Jussiani Moreira e João Victor Laskoski

Versão: 31-Ago-2022

1 Introdução

O presente relatório refere-se a prática de laboratório 3 da disciplina, em que o laboratório trata de uma rotina que ao receber uma determinada imagem em tons de cinza, deve gerar um histograma, mapeando a quantidade de pixels para cada valor da escala de “luminosidade”, sendo de 0 até 255.

Além disso, tal rotina deve ser implementada em assembly, porém sendo chamada por um programa implementado em C++. Sendo assim, tal prática pretende trazer maior aprofundamento nos comandos em Assembly, salientando o desempenho da rotina, utilizados por diversos sistemas embarcados.

2 Planejamento das fases do processo de desenvolvimento

- Estudo do código images.c.
- Verificação das especificações e padronização dos componentes utilizados no projeto (funções e arquivos).
- Configurações básicas do projeto.
- Verificação de como criar e executar uma rotina Assembly por um programa em C++.
- Estudo do algoritmo para criação do histograma
- Verificação de comandos Assembly úteis e elaboração da rotina em Assembly
 - Implementação e testes de enviar as variáveis por parâmetro e receber corretamente na rotina em Assembly
 - Implementação e testes de inicializar o vetor do histograma na rotina em Assembly
 - Implementação e testes de popular o vetor do histograma, como também retornar corretamente para o programa em C++ as variáveis necessárias
- Elaboração dos casos de teste 1 e 2 e execução dos mesmos

3 Definição do problema a ser resolvido

Sendo assim, a prática em questão tem como propósito gerar um histograma, o qual contém a relação a quantidade de pixels para cada valor na escala de tons de cinza. Ainda mais, tal prática deve ser executada por uma rotina desenvolvida em Assembly.

4 Especificação da solução

Logo, a solução situa-se da seguinte forma, tendo previamente o vetor da imagem e parâmetros necessários como altura e largura da imagem que estão contidos no arquivo images.c, os mesmos serão enviados com adição de um vetor que representa o histograma para uma rotina implementada em Assembly, a qual deverá ser importada pelo programa em C++.

Tal rotina deverá verificar primeiramente o tamanho da imagem, caso seja maior que um tamanho limite, deverá retornar erro. Caso contrário, será inicializado o vetor do histograma (em que inicialmente está com dados nulos). Consequente, será percorrido o vetor da imagem, verificando o valor da escala de cinza de cada pixel, tal valor será mapeado com a posição do histograma e será adicionado que há um pixel para tal valor da escala no vetor do histograma.

Ainda mais, será retornado o número de bits processados e pelo programa em C++ será gerado testes, verificando o vetor de histograma modificado e se os resultados estão

coerentes com o esperado.

Sendo assim, as restrições impostas estão no formato da declaração da função, como também o tamanho máximo da imagem que a rotina em Assembly irá tratar e o tamanho do vetor do histograma. E na questão da funcionalidade, está a contagem da quantidade de pixels que possui determinado valor de “luminosidade”.

5 Estudo da plataforma de HW

Para esta prática, como não foram utilizados periféricos, temporizadores ou interrupção, não foi necessário o estudo dessas funcionalidades relacionando com a Tiva. Visto que, para essa prática, será apenas utilizado o microcontrolador para executar um programa em C++ executando uma rotina em Assembly e o resultado obtido será verificado no terminal da IDE.

6 Projeto (design) da solução

A solução foi construída obedecendo o fluxograma abaixo.

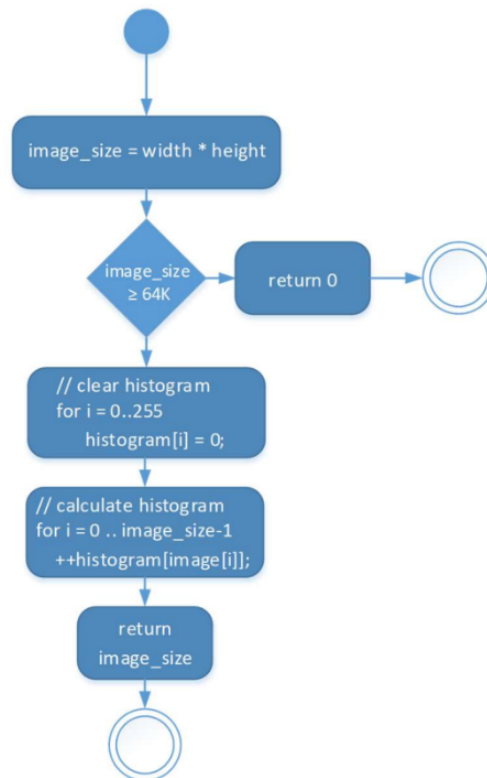


Figura 1 - Fluxograma.

Para a prática em questão, não será necessário a utilização de funções da biblioteca TivaWare, sendo utilizado apenas funções básicas do C++ (para loop, chamada de funções e print no terminal), como também comandos em Assembly.

Contudo, para a rotina em Assembly, serão utilizados possivelmente os seguintes comandos:

- LDR(B | H) : Carregar da memória uma variável (byte ou half-word)
- STR(B | H) : Escrever na memória uma variável (byte ou half-word)
- MOV: Mover um valor para um reg. destino
- CMP: Comparação de valores para verificação de flags
- B(condição) <label>: Pulo condicional para o label especificado
- BX LR: Retorna para o endereço apontado por Link Register
- ADD: operação de soma
- SUB: operação de subtração

7 Configuração do projeto na IDE (IAR).

Não foi utilizada nenhuma configuração diferente da configuração padrão obtida com a criação do projeto.

8 Teste e depuração.

Os testes serão divididos em partes, verificando se cada bloco da rotina em Assembly, como também a chamada e retorno pelo programa em C++ serão executadas devidamente. Os testes serão feitos da seguinte forma:

- Carregamento da matriz de imagens do images.c sendo feita corretamente, verificando os valores através de prints no terminal da IDE.
- Recebimento correto dos parâmetros da função pela rotina Assembly, em que é verificado através dos valores dos registradores (R0 ~ R3).
- Inicialização do vetor de histogramas, em que o teste recai sobre verificar o painel de memórias, vendo se as devidas posições estão zeradas.
- População do vetor de histograma, em que pode ser verificado através dos testes 1 e 2, printando os valores de todas as posições do histograma e comparando com um gráfico gerado em Excel.