Lab 2 – Bruno Guillen

1. Planejamento das fases do processo de desenvolvimento.

- Entendimento do Histograma
- Estudo do processador ARM Cortex-M4F
- Entendimento do Problema a ser resolvido
- Compreender a especificação dada

2. Definição do problema a ser resolvido.

Elaborar uma rotina em assembly que será chamada de um programa em C++. A rotina deve gerar um histograma de uma imagem em tons de cinza.

3. Especificação da solução.

Requisitos funcionais:

RF1 - O programa deve calcular o histograma de uma imagem.

RF1.1 - o Tamanho da imagem deve ser cedido no código.

RF2 - O programa deve apresentar o tamanho da imagem no terminal do IAR.

RF3 - O programa deve apresentar o histograma no Terminal do IAR.

RF4 – O programa dará resposta 0 a valores maiores que 64k.

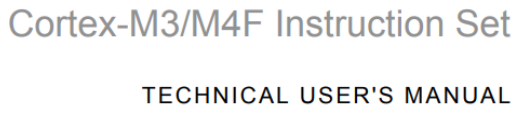
Restrições:

- usar funções devem ser chamadas por um codigo em C++

- a solução deve fazer uso de assembly.

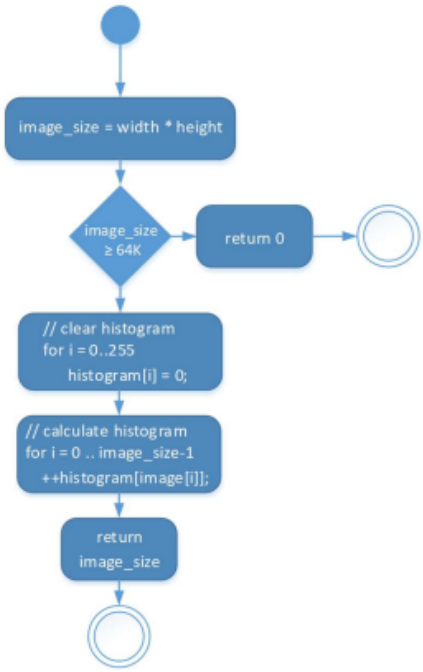
4. Estudo da plataforma de HW (placa Tiva e seu processador).

Foi pesquisado as instruções assembly padroes para que então fossem consultadas e adaptadas de acordo com o manual de instruções do processador da placa.



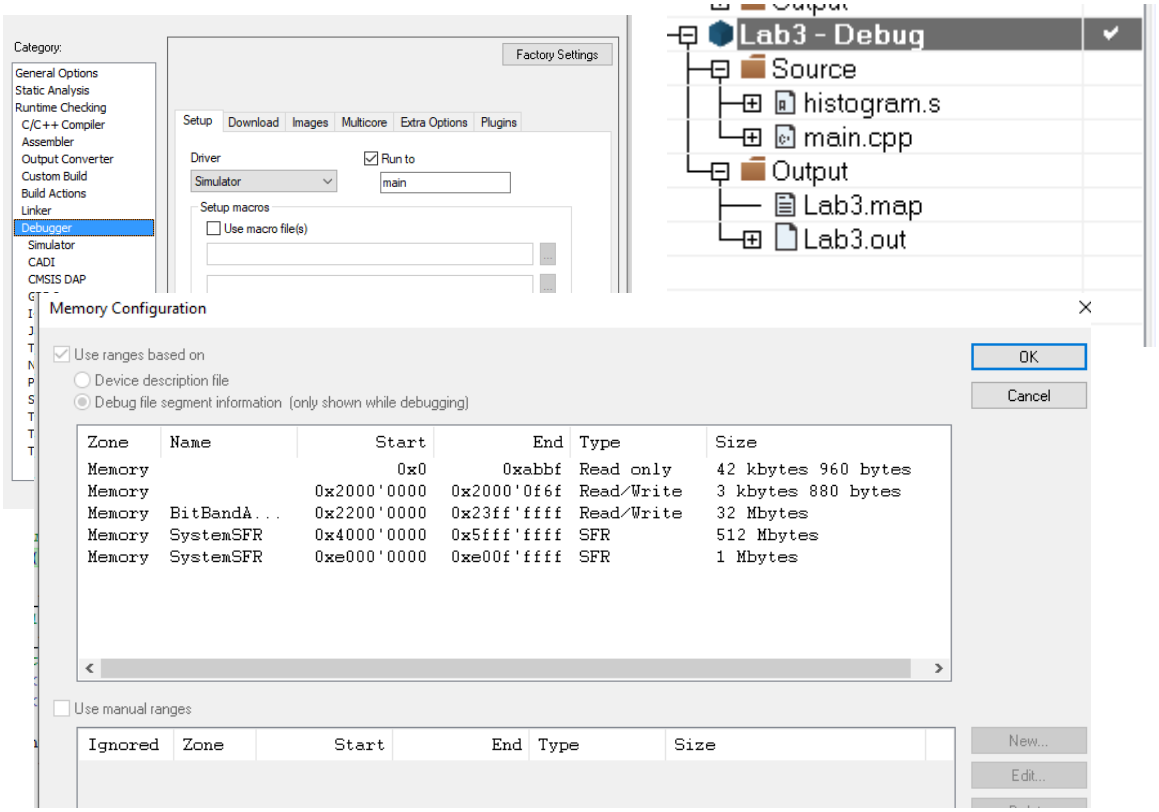
6.Projeto (design) da solução:

A solução foi desenvolvida em cima da solução proposta pela atividade



7. Configuração do projeto na IDE (IAR).

Estrutura e Configuração:



8. Edição do código da solução.

Estrutura em C:
Adicionada a importação da função em assembly
extern "C" uint16_t EightBitHistogram(uint16_t width, uint16_t height, uint8_t * p_image, uint16_t * p_histogram);
e valores para imagem de teste0

```
const uint8_t image0[HEIGTH0][WIDTH0] = {
    { 10, 100, 5, 55}, {255, 0, 255, 0},{ 10, 100,75, 75} };
```

Inicialização do vetor para histograma e chamada da função:

```
int main()
{
  uint16_t vector[256] = {0}; //initialize vector
 // uint16_t size = EightBitHistogram(WIDTH0, HEIGTH0, (uint8_t*) *image0, vector);
  uint16_t size = EightBitHistogram(WIDTH1, HEIGTH1, (uint8_t*) *image1, vector); //recieves size
```

Prints necessárias: para os requisitos levantados

```
//printing stuffs
cout << "Image size:" << size << '\n';
cout << "Histogram: \n" ;
for (int i=0;i<256;i++){
  cout << vector[i] << ',';
}
```

Posições importantes salvas no assembly

```
EightBitHistogram
    MOV R9, R3 ;store initial position (R9)
    MUL R6, R0, R1 ;store size (R6)
```

Comparativo para atender o tamanho da imagem retornando zero:

```
ITT GT ;if >64k
  MOVGT R0, #0
  BXGT LR ;return 0
```

Adicionando e Removendo das pilhas para as chamadas de funções:

```
PUSH {LR} ;put on stack
BL clearHistogram
POP {LR} ;clear  stack
```

```
PUSH {LR} ;put on stack
BL calculateHistogram
POP {LR} ;clear  stack
```

Loop para limpeza do histograma utilizando seus offsets.

```
clearHistogram
  STR R8, [R3], #2 ;zeros the actual position and it`s offset
  ADD R7, #1
  CMP R7, #256 ;see if its the end
  BNE clearHistogram ;loop
  BX LR
```

Inicia-se na posição inicial do histograma

```
calculateHistogram
  MOV R3, R9 ;get start position
```

Le se o valor na posição atual do vetor para pegar seu valor

```
LDR R11, [R3, R8] ;read the value of the actual position
```

Para então atualizar no valor correto do vetor.

```
STR R11, [R3] ;update the vector value position
```

e comparasse com a posição final Definida como R6 para permanecer no loop

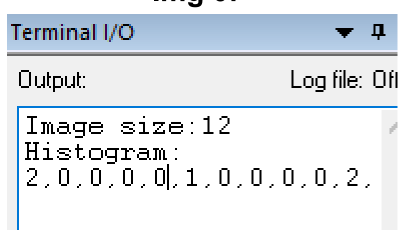
```
ADD R7, #1
CMP R7, R6  ;see if its the end
BNE calculateHistogram ;loop
BX LR
```

Como se trata de uma Função para 8bits e o seu retorno é em 16 é necessárias a conversão o que no caso ocorre quando se dobra o valor de R8

```
ADD R8, R8 ; double the value
```

9. Teste e depuração.

Img 0:



2,0,0,0,0,1,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2

IMG1:

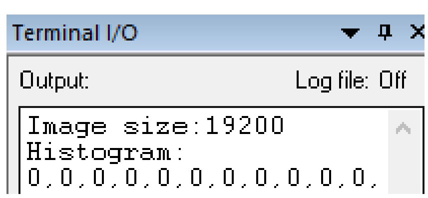


Image size:19200
Histogram:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,12,15,16,26,5,402,32,9,9,20,18,2,13,16,6,12,15,6,1,2,6,4,2,10,11,7,5,8,15,
12,16,13,3,16,28,31,48,50,53,3,50,48,25,21,25,27,0,17,21,25,23,20,24,2,21,15,23,20,32,21,18,4,15,2
6,30,18,24,26,1,29,57,63,86,114,113,24,180,150,169,183,177,179,13,218,185,160,187,179,192,14,13
9,161,131,120,83,84,8,44,52,65,60,75,64,4,69,66,71,69,96,86,13,88,92,78,111,136,123,16,119,131,1

76,161,195,187,31,222,260,298,265,244,255,38,184,270,272,282,286,325,373,54,267,296,292,311,240,303,43,248,216,216,264,338,258,29,268,257,255,258,283,290,32,198,296,232,210,107,93,11,65,63,73,48,37,42,12,34,28,36,32,29,34,5,41,47,37,38,53,49,5,46,49,42,38,29,33,2,33,39,42,49,72,96,0,80,40,32,34,22,29,29,2,19,24,22,17,16,16,0,13,31,38,57,36,14,12,3,4,2,1,2,1,3,0,0,0,0,1,0,0,4,1,2])

Comparando as imagens se obteve um resultado satisfatório