

Computação Heterogênea de Alto Desempenho - Lab 7

Manuel Santos - 2019231352

12 Dezembro 2023

Introdução

O presente relatório descreve a resolução dos exercícios presentes na ficha 7 de Computação Heterogênea de Alto Desempenho, que abordam o *OpenCL*.

Exercício 1

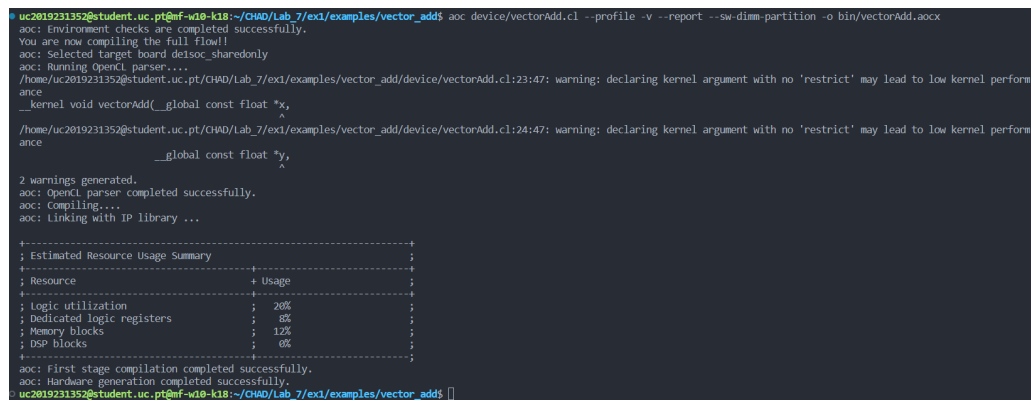
No exercício 1, segui as indicações e explicações dadas para correr com sucesso o código disponibilizado.

O *kernel* utilizado neste exercício foi o que se segue:

```
__kernel void vectorAdd(__global const float *x,
                        __global const float *y,
                        __global float *restrict z)
{
    // get index of the work item
    int index = get_global_id(0);

    // add the vector elements
    z[index] = x[index] + y[index];
}
```

De forma a testar este *kernel*, efetuei a devida sintetização e compilação, cujos resultados estão visíveis nas figuras 1 e 2 respetivamente.



```
uc2019231352@student.uc.pt@mfw10-k18:~/CHAD/Lab_7/ex1/examples/vector_add$ aoc device/vectorAdd.cl --profile -v --report --sw-dimm-partition -o bin/vectorAdd.aocx
aoc: Environment checks are completed successfully.
You are now compiling the full flow!
aoc: Selected target board de1soc_sharedonly
aoc: Running OpenCL parser...
/home/uc2019231352@student.uc.pt/CHAD/Lab_7/ex1/examples/vector_add/device/vectorAdd.cl:123:47: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void vectorAdd(__global const float *x,
                        ^
/home/uc2019231352@student.uc.pt/CHAD/Lab_7/ex1/examples/vector_add/device/vectorAdd.cl:124:47: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
                        __global const float *y,
                        ^
2 warnings generated.
aoc: OpenCL parser completed successfully.
aoc: Compiling...
aoc: Linking with IP library ...

+-----+-----+
; Estimated Resource Usage Summary
+-----+-----+
; Resource                               ; Usage
+-----+-----+
; Logic utilization                       ; 26%
; Dedicated logic registers               ; 8%
; Memory blocks                           ; 12%
; DSP blocks                             ; 6%
+-----+-----+
aoc: First stage compilation completed successfully.
aoc: Hardware generation completed successfully.
uc2019231352@student.uc.pt@mfw10-k18:~/CHAD/Lab_7/ex1/examples/vector_add$
```

Figura 1: Resultado da sintetização

```

root@f-v10-k18:~/CWD/Lab_7/ex1/examples/vector_add# make
arm-linux-gnueabihf-g++ host/src/main.cpp ../common/src/aocl_utils.cpp -o vector_add -I/usr/local/altera/14.1/hld/host/include -I../common/inc -I/usr/local/altera/14.1/hld/board/deisoc/arm32/lib -I/usr/local/altera/14.1/hld/host/arm32/lib -I/usr/local/altera/14.1/hld/host/arm32/lib -lalteracli -laci_emulator_kernel_rt -lalterahalmid -lalterammpcie -lert -ldl -lstdc++
root@f-v10-k18:~/CWD/Lab_7/ex1/examples/vector_add#

```

Figura 2: Resultado da compilação

Por fim, corri o programa, estando o resultado do mesmo visível na figura 3.

```

File Edit View Bookmarks Settings Help
root@socfpga:~/vector_Add# ls
profile.mon  vectorAdd  vectorAdd.aocx  vector_add
root@socfpga:~/vector_Add# aocl program /dev /acl0 vectorAdd.aocx
aocl program: Running reprogram from /home/root/openc1_arm32_rte/board/c5soc/arm32/bin
aocl program - Configures a new FPGA design onto your board

Usage: aocl program <device_name> <file.aocx>

Supply the .aocx file for the design you wish to configure onto
the FPGA. You need to provide <device_name> to specify the FPGA
device to configure with.

Description:

This command downloads a new design onto your FPGA.
This utility should not normally be used, users should instead use
c1CreateProgramWithBinary to configure the FPGA with the .aocx file.

root@socfpga:~/vector_Add# aocl program /dev/acl0 vectorAdd.aocx
aocl program: Running reprogram from /home/root/openc1_arm32_rte/board/c5soc/arm32/bin
Reprogramming was successful!
root@socfpga:~/vector_Add# ./vectorAdd
Initializing OpenCL
Platform: Altera SDK for OpenCL
Using 1 device(s)
  deisoc_sharedonly : Cyclone V SoC Development Kit
Using AOXC: vectorAdd.aocx
Launching for device 0 (1000000 elements)

Time: 115.998 ms
Kernel time (device 0): 6.819 ms

Verification: PASS
root@socfpga:~/vector_Add#

```

Figura 3: Resultado da execução

Exercício 2

No exercício 2, segui os exemplos disponíveis de forma a obter um programa capaz de multiplicar 2 matrizes. Escolhi fazer com matrizes e não vetores de forma a servir de apoio à resolução do exercício 3.

O *kernel* utilizado neste exercício foi o que se segue:

```

__kernel void matMul(__global const float *A,
                    __global const float *B,
                    __global float *restrict C)
{
    // == Get the row and column of the element
    int row = get_global_id(1);
    int col = get_global_id(0);

    // == Check if the row and column are within the matrix bounds
    if ((row < 1024) && (col < 1024))
    {
        float sum = 0.0;
        for (int k = 0; k < 1024; k++)
            sum += A[row * 1024 + k] * B[k * 1024 + col];
        C[row * 1024 + col] = sum;
    }
}

```

De forma a testar este *kernel*, efetuei a devida sintetização e compilação, cujos resultados estão visíveis nas figuras 4 e 5 respetivamente.

```

root@mf-w10-k18:~/CHAD/Lab_7/ex2# aoc device/matMul.cl --profile -v --report --sw-dimm-partition -o bin/matMul.aocx
aoc: Environment checks are completed successfully.
You are now compiling the full flow!!
aoc: Selected target board deisoc_sharedonly
aoc: Running Opencl parser...
/home/uc2019231352@student.uc.pt/CHAD/Lab_7/ex2/device/matMul.cl:15:44: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void matMul(__global const float *A, __global const float *B, __global float *restrict C)
                                         ^
/home/uc2019231352@student.uc.pt/CHAD/Lab_7/ex2/device/matMul.cl:15:69: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void matMul(__global const float *A, __global const float *B, __global float *restrict C)
                                                                    ^
2 warnings generated.
aoc: Opencl parser completed successfully.
aoc: Compiling...
aoc: Linking with IP library ...

+-----+
; Estimated Resource Usage Summary
+-----+
; Resource                               + Usage
+-----+
; Logic utilization                       ; 31%
; Dedicated logic registers               ; 13%
; Memory blocks                          ; 25%
; DSP blocks                             ; 1%
+-----+
aoc: First stage compilation completed successfully.
aoc: Hardware generation completed successfully.
root@mf-w10-k18:~/CHAD/Lab_7/ex2#

```

Figura 4: Resultado da sintetização

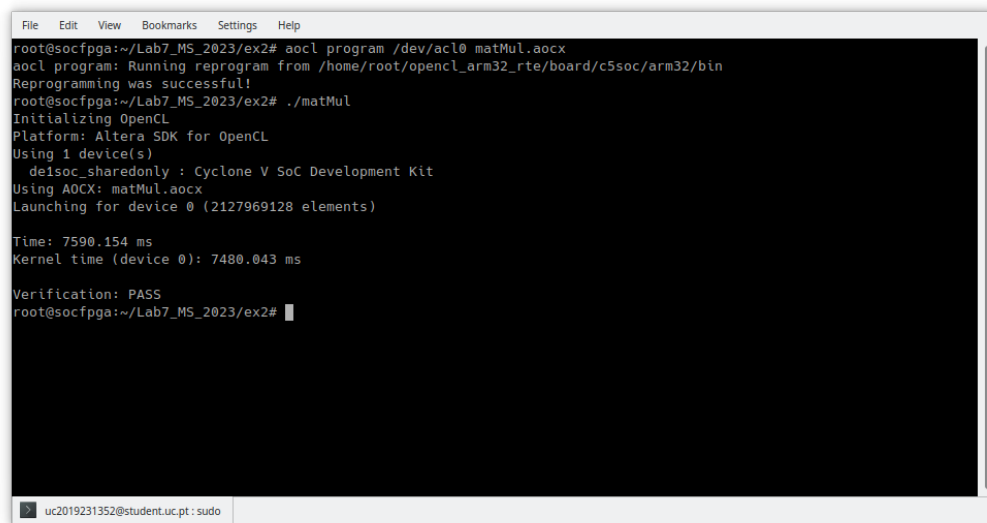
```

root@mf-w10-k18:~/CHAD/Lab_7/ex2# make
arm-linux-gnueabihf-g++ host/src/main.cpp ../common/src/AOCI_utils.cpp -o matMul -I/usr/local/altera/14.1/hld/host/include -I../common/inc -I/usr/local/altera/14.1/hld/board/deisoc/arm32/lib -I/usr/local/altera/14.1/hld/host/arm32/lib -I/usr/local/altera/14.1/hld/host/arm32/lib -lalteracl -laci_emulator_kernel_rt -lalterahalmnd -lalteramdpic -lelf -lrt -ldl -lstdc++
root@mf-w10-k18:~/CHAD/Lab_7/ex2#

```

Figura 5: Resultado da compilação

Por fim, corri o programa, estando o resultado do mesmo visível na figura 6.



```
File Edit View Bookmarks Settings Help
root@socfpga:~/Lab7_MS_2023/ex2# aocl program /dev/acl0 matMul.aocx
aocl program: Running reprogram from /home/root/opencl_arm32_rte/board/c5soc/arm32/bin
Reprogramming was successful!
root@socfpga:~/Lab7_MS_2023/ex2# ./matMul
Initializing OpenCL
Platform: Altera SDK for OpenCL
Using 1 device(s)
  de1soc_sharedonly : Cyclone V SoC Development Kit
Using AOCLX: matMul.aocx
Launching for device 0 (2127969128 elements)

Time: 7590.154 ms
Kernel time (device 0): 7480.043 ms

Verification: PASS
root@socfpga:~/Lab7_MS_2023/ex2#
```

Figura 6: Resultado da execução

A execução deste programa demorou bastante tempo, como é possível verificar na figura 6. Como tal, existe bastante espaço para a melhoria do *kernel*.

Exercício 3

No exercício 3, utilizei os primeiros exercícios como base para a minha resolução, em especial a parte do código do *host*.

O *kernel* que desenvolvi para a resolução desde exercício foi o que se segue:

```
__kernel void vectorAdd(__global const float *x,
                        __global const float *y,
                        __global float *restrict z)
{
    // get index of the work item
    int index = get_global_id(0);

    // add the vector elements
    z[index] = x[index] + y[index];
}
```

De forma a testar este *kernel*, efetuei a devida sintetização e compilação, cujos resultados estão visíveis nas figuras 7 e 8 respetivamente.

```

uc2019231352@student.uc.pt@mf-w10-k18:~/CHAD/Lab_7/ex3$ aoc device/GreyScale.cl --profile -v --report --sw-dimm-partition -o bin/GreyScale.aocx
aoc: Environment checks are completed successfully.
You are now compiling the full flow!
aoc: Selected target board deisoc_sharedonly
aoc: Running OpenCL parser....
/home/uc2019231352@student.uc.pt/CHAD/Lab_7/ex3/device/GreyScale.cl:10:56: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void GreyScale(__global const unsigned char * rgbImage,
                        ^
/home/uc2019231352@student.uc.pt/CHAD/Lab_7/ex3/device/GreyScale.cl:11:32: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
                        __global unsigned char * grayImage)
                        ^
2 warnings generated.
aoc: OpenCL parser completed successfully.
aoc: Compiling....
aoc: Linking with IP library ...

+-----+
; Estimated Resource Usage Summary
+-----+
; Resource      + Usage
+-----+
; Logic utilization      ; 21%
; Dedicated logic registers ; 9%
; Memory blocks          ; 10%
; DSP blocks             ; 3%
+-----+
aoc: First stage compilation completed successfully.
aoc: Hardware generation completed successfully.
uc2019231352@student.uc.pt@mf-w10-k18:~/CHAD/Lab_7/ex3$

```

Figura 7: Resultado da sintetização

```

root@mf-w10-k18:~/CHAD/Lab_7/ex3$ make
arm-linux-gnueabihf-g++ -g host/src/main.cpp ../common/src/AOCL_utils.cpp -o GreyScale -I/usr/local/altera/14.1/hld/host/include -I../common/inc -I/usr/local/altera/14.1/hld//board/deisoc/arm32/lib -I/usr/local/altera/14.1/hld/host/arm32/lib -I/usr/local/altera/14.1/hld/host/arm32/lib -lalteracl -laci_emulator_kernel_rt -lalterahalmc -lalteramdpic
root@mf-w10-k18:~/CHAD/Lab_7/ex3$

```

Figura 8: Resultado da compilação

Por fim, tentei correr o programa, sem sucesso. Tentei várias vezes depurar o código do *host*, visto que acredito que o código do *kernel* está correto.

O maior problema que identifiquei prende-se na transferência dos dados do *host* para o *device*, que por algum motivo que ainda não compreendi, não era bem sucedido.

Conclusão

Apesar de não ter sido completamente bem sucedido, a resolução destes exercícios serviram como uma boa forma de experimentar e pôr em prático os conceitos de *OpenCl* adquiridos.