

Lab 6 - OpenCL em FPGA

Marta Nunes, nº2017246232

Neste trabalho pretendemos implementar 2 exercícios em OpenCL na FPGA. Uma vez que não foi possível executar os exemplos na FPGA vou apresentar a compilação do código implementado por mim.

1 Exercício 1 - Multiplicação de dois arrays

Neste exercício queremos executar a multiplicação de 2 arrays. Para a implementação do mesmo considere (por uma questão de simplicidade, tal como fazíamos em CUDA) que tinha 2 matrizes quadradas com as dimensões SIZExSIZE, sendo que estas matrizes são passadas em forma de array com uma dimensão e tamanho $N=SIZE*SIZE$ para a kernel. Dentro desta temos o código apresentado de seguida.

```
// ACL kernel for multiply two input vectors
__kernel void vectorMul(__global const float *x,
                        __global const float *y,
                        __global float *restrict z)
{
    // get index of the work item
    int index = get_global_id(0);

    // multiply the vector elements
    int i = index / SIZE;
    int j = index % SIZE;
    float value;

    if (i < SIZE)
    {
        value = 0.0;
        for (int k=0; k<SIZE; k++)
            value += x[i*SIZE + k] * y[k*SIZE + j];
        z[i*SIZE + j] = value;
    }
}
```

```

    }
}

```

Neste código começo por a partir do índice global calcular a linha e a coluna do pixel a calcular e usando essa informação vou buscar os valores da linha da primeira matriz e da coluna da segunda matriz que necessito para realizar a operação de multiplicação. A compilação do código resultou no seguinte.

```

uc2017246232@student.uc.pt@mf-w10-k18:~/lab6/vector_mul$ aoc device/vectorMul.cl --profile -v --report --sw-dump-partition -o bin/vectorAdd.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/uc2017246232@student.uc.pt/lab6/vector_mul/device/vectorMul.cl:2:47: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void vectorMul(__global const float *x,
                        ^
/home/uc2017246232@student.uc.pt/lab6/vector_mul/device/vectorMul.cl:3:47: 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                       ; 29%
; Dedicated logic registers               ; 13%
; Memory blocks                           ; 19%
; DSP blocks                              ; 3%
+-----+
aoc: First stage compilation completed successfully.

```

Figure 1: Resultado da compilação da multiplicação de matrizes.

No ficheiro main.cpp mantive o código de exemplo dado, fazendo as alterações necessárias para o exercício, como definição do tamanho da matriz e alteração do código do cálculo da saída de referência.

2 Exercício 2 - Conversão de imagem RGB para grayscale

Neste exercício queremos converter uma imagem RGB para grayscale usando a seguinte fórmula para cada pixel.

$$grayPixel[i, j] = 0.21 * r + 0.71 * g + 0.07 * b \quad (1)$$

Para isso implementei o seguinte código da kernel.

```

// ACL kernel for convert a image to grayscale
__kernel void colorToGrey(__global const float *x,
                          __global float *restrict z)
{
    // get index of the work item
    int index = get_global_id(0);

    // multiply the vector elements

```

```

float value;

if (index < SIZE)
{
    int offset = index * 3;
    z[index] = 0.21*x[offset] + 0.71*x[offset + 1] + 0.07*x[offset + 2];
}
}

```

A compilação deu o seguinte resultado.

```

uc2017246232@student.uc.pt@mf-w10-k18:~/lab6/conversao$ aoc device/colorToGrey.cl --profile -v --report --sw-dmm-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/uc2017246232@student.uc.pt/lab6/conversao/device/colorToGrey.cl:2:49: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void colorToGrey(__global const float *x,
                        ^
1 warning generated.
aoc: OpenCL parser completed successfully.
aoc: Compiling....
aoc: Linking with IP library ...

+-----+
; Estimated Resource Usage Summary
+-----+
; Resource           + Usage
+-----+
; Logic utilization      ; 22%
; Dedicated logic registers ; 10%
; Memory blocks         ; 11%
; DSP blocks            ; 11%
+-----+
aoc: First stage compilation completed successfully.

```

Figure 2: Resultado da compilação da conversão de uma imagem.

Por uma questão de simplicidade, e tal como já tinha feito anteriormente em CUDA, neste exercício inicializei uma matriz com valores aleatórios em vez de ler uma imagem. Mencionar ainda que, tal como efetuei no exercício anterior, também neste exercício passei para dentro do kernel o vetor que representa a imagem à qual a conversão irá ser realizada. Assim sendo, basta então dentro do kernel aplicar a fórmula dada, tendo em conta que a imagem RGB tem o triplo do tamanho da imagem grayscale. Esta diferença de tamanhos vai também influenciar o ficheiro main.cpp, que foi adaptado para este problema.