1) Describe el pipeline gráfico.

2) Dispones de 850 euros para comprar una tarjeta gráfica. ¿Cuál comprarías? Justifica la

respuesta. Has de buscar dos tarjetas: una para jugar y otra para un entorno de computación

de altas prestaciones.

3) Si queremos utilizar GPUs para cálculo de propósito general (GPGPU) puedes escoger entre

CUDA, OpenCL y OpenACC. Describe las ventajas e inconvenientes de cada alternativa.

Además, se pueden combinar. ¿Qué posibilidades ofrece esta alternativa?

4) Dada la siguiente rutina escrita en C:

void Examen18(float mA[N][M], float mB[N][M], float vC[N], float vD[M])

{int i, j;

for (i=0; i<N; i++)

for (j=0; j<M; j++)

mA[i][j] = mA[i][j]\*vC[j] - mB[i][j]\*vD[i] + mA[i][0]\*mB[8][j]; }

Escribid 3 versiones del kernel CUDA que resuelva el mismo problema:

a) En la primera versión cada thread se va a ocupar de 1 columna de la matriz resultado.

b) En la segunda versión cada thread se va a ocupar de 1 fila de la matriz resultado.

c) En la última versión cada thread se va a ocupar de 1 elemento de la matriz resultado.

Escribid los kernels CUDA para cada versión, así como la invocación correspondiente. Tened

en cuenta que como máximo podéis utilizar 1024 threads por bloque y que las variables N y

M pueden tener cualquier valor (p.e. N = 1237, M = 2311).

5) Hablando de texturas, ¿qué filtros existen?, ¿puedes describirlos? ¿qué implicaciones tienen?

6) ¿Qué es nvprof? ¿Qué información da? ¿Para qué sirve? Enumera y explica las que consideres más importantes

nvprof is a command-line profiler that enables you to collect and view CPU and GPU timers and events in CUDA programs.

nvprof es la herramienta que nos proporciona NVIDIA para hacer profiling de nuestras aplicaciones

CUDA. Junto con los ficheros de la sesión os hemos dejado una copia del man del comando nvprof

(fichero help.nvprof).

Sería muy interesante que lanzarais alguna de las ejecuciones añadiendo alguno de estos flags al

nvprof:

• --print-gpu-summary, genera un sumario de tiempos de los kernels y transferencias

• --print-gpu-trace, genera una traza de los kernels y transferencias

• --metrics all, para cada kernel muestra una serie de métricas de rendimiento (las genera

TODAS, y hay 124 métricas diferentes). Las métricas disponibles dependen de la GPU.

• Podemos especificar las métricas que queremos por separado. Algunas métricas interesantes

son las siguientes:

o sm\_efficiency

o achieved\_occupancy

o gld\_requested\_throughput

o gst\_requested\_throughput

o dram\_utilization

<http://fisica.cab.cnea.gov.ar/gpgpu/images/2016/clases/02_profiling.pdf>

<http://www.mastergraficos.com/wp/wp-content/uploads/2015/06/4a_PGATR.pdf>

8) En los próximos años es muy posible que la mayoría de las tarjetas gráficas de alto

rendimiento utilicen HBM (High Bandwidth Memory). Explica las características de la memoria

HBM y las ventajas que ofrece respecto a una memoria GDDR5 convencional.

<https://www.youtube.com/watch?v=50Yw7AM1OKg>

<https://www.eurogamer.es/articles/digitalfoundry-2015-amd-tecnologia-hbm>

9) Describe las directivas más importantes de OpenACC. Explícalas incluyendo un ejemplo de

uso.

10)En la invocación de un kernel de CUDA se pueden utilizar hasta 4 parámetros:

KernelCUDA<<<par1, par2, par3, par4>>>(…)

Déscribelos y pon ejemplos de utilización de los mismos.

Dimensiones de la grilla (2D): variable de tipo dim3 • Dimensiones de los bloques de threads (3D): variable de tipo dim3 • Opcionales: • cantidad de memoria compartida por bloque • identificador de stream

<https://www.fing.edu.uy/inco/cursos/gpgpu/clases/C4PC2x.pdf>