УДК 43
Грондзаль А. – ст. гр. СІ-21
Тернопільський національний технічний університет імені Івана Пулюя

ANALYSIS OF THE COMPUTER SYSTEM BASED ON NVIDIA FERMI ARCHITECTURE

Науковий керівник: Перенчук О.З.

A computer is a programmable machine designed carry out a sequence of arithmetic or logical operations automatically. Nowadays a computer is an integral part of academic and social life and indicator of progress. I think it will continue in the nearest future. Therefore, profession of a computer engineer is very important nowadays.

Computer engineering, also called computer systems engineering, is a discipline that integrates several fields of electrical engineering and computer science required to develop computer systems.

Most of computer systems is personal ones and one of its main components is a video card. Therefore, review the latest architectures is a relevant topic.

The graphics processing unit (GPU), first invented by NVIDIA in 1999, is the most common parallel processor now. Fuelled by the insatiable desire for life-like real-time graphics, the GPU has evolved into a processor with unprecedented floating-point performance and programmability; today's GPUs greatly outpace CPUs in arithmetic throughput and memory bandwidth, making them the ideal processor to accelerate a variety of data parallel applications.

The Fermi architecture is the most significant leap forward in GPU architecture since the original G80 architecture. G80 was our initial vision of what a unified graphics and computing parallel processor should look like. GT200 extended the performance and functionality of G80. With Fermi, we have taken all we have learned from the two prior processors and all the applications that were written for them, and employed a completely new approach to design to create the world's first computational GPU.

The Fermi architecture consists of CUDA cores. CUDA is the hardware and software architecture that enables NVIDIA GPUs to execute programs written with C, C++, Fortran, OpenCL, DirectCompute, and other languages. A CUDA program calls parallel kernels. A kernel executes in parallel across a set of parallel threads. The programmer or compiler organizes these threads in thread blocks and grids of thread blocks. The GPU instantiates a kernel program on a grid of parallel thread blocks. Each thread within a thread block executes an instance of the kernel, and has a thread ID within its thread block, program counter, registers, per-thread private memory, inputs, and output results.

A thread block is a set of concurrently executing threads that can cooperate among themselves through barrier synchronization and shared memory. A grid is an array of thread blocks that execute the same kernel, read inputs from global memory, write results to global memory, and synchronize between dependent kernel calls.

These innovations require detailed study. Our task as future computer engineers to review this subject which is an urgent problem nowadays.