

На рисунке 1 представлено окно плоского шлифование торцом круга на станках с круглым столом.

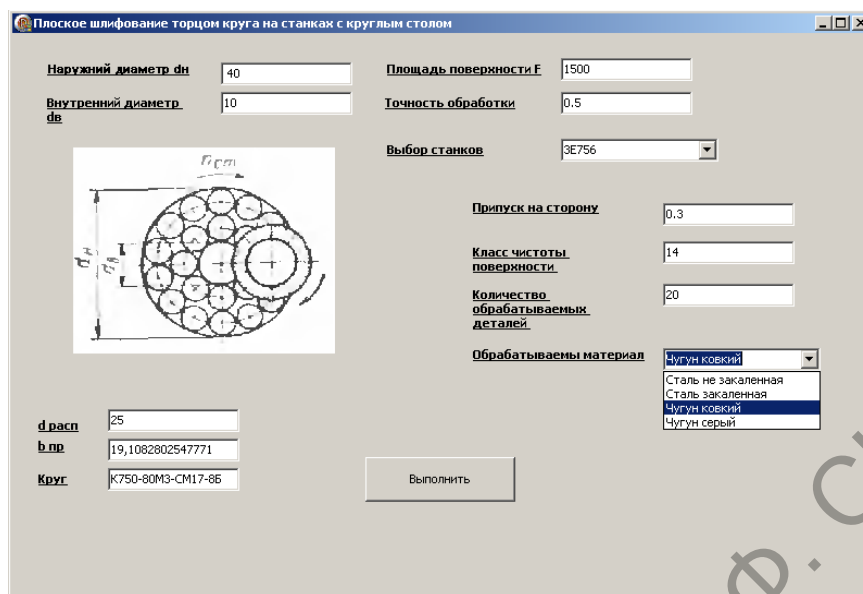


Рисунок 1 – Плоское шлифование торцом круга на станках с круглым столом

Основным результатом разработанной программы является инструкционная карта, в которой указано наименование операции, условия обработки, сведения о станке и характеристика шлифовального круга, оптимальные режимы резания, а также основное технологическое (машинное) время, необходимое для выполнения операции.

### Литература

1. Режимы резания металлов. Справочник. Под ред. Ю.В. Барановского. – М.: Машиностроение, 1972. – 408 с.

**Я.А. Юницкий (УО «ГГУ им. Ф.Скорины», Гомель)**

Науч. рук. **А.В. Воружев**, канд. техн. наук, доцент

### ПРИМЕНЕНИЕ ТЕХНОЛОГИИ NVIDIA CUDA ДЛЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Устройства для превращения персональных компьютеров в миниатюрные суперкомпьютеры известны довольно давно. В 80-х гг. прошлого века на рынке они предлагались под названием «транспьютеры». Но в последнее время эстафета параллельных вычислений перешла к массовому рынку, так или иначе связанному с трёхмерными играми.

Графическая составляющая современных приложений неуклонно стремиться к совершенству – фотореализму. Но для столь сложного рендера в реальном времени необходимо максимальное быстродействие вычислительного устройства. Графические адаптеры с многоядерными процессорами для параллельных векторных вычислений, используемых в 3D-графике, достигают высокой пиковой производительности, которая универсальным процессорам не под силу.

В конечном итоге случилось то, что должно было – графические адаптеры стали применяться для неграфических расчетов. Современные видеочипы содержат сотни математических исполнительных блоков, и вся эта мощь может быть использована для ускорения различных «сильных» приложений.

Компания NVIDIA в 2007 г. представила технологию CUDA. Технология CUDA – это программно-аппаратная вычислительная архитектура NVIDIA, основанная на расширении языка Си, которая даёт возможность организации доступа к набору инструкций графического ускорителя и управления его памятью при организации параллельных вычислений. В основе API лежит расширенный язык Си, а для трансляции кода с этого языка в состав CUDA SDK входит собственный компилятор.

Основные характеристики CUDA:

- унифицированное программно-аппаратное решение для параллельных вычислений на видеочипах NVIDIA;
- большой набор поддерживаемых решений;
- стандартный язык программирования Си;
- стандартные библиотеки численного анализа FFT (быстрое преобразование Фурье) и BLAS (линейная алгебра);
- оптимизированный обмен данными между CPU и GPU;
- взаимодействие с графическими API OpenGL и DirectX;
- возможность разработки на низком уровне.

Преимущества CUDA:

- интерфейс программирования приложений CUDA основан на стандартном языке программирования Си с расширениями;
- CUDA обеспечивает доступ к разделяемой между потоками памяти размером в 16 Кб на мультипроцессор, которая может быть использована для организации кэша с широкой полосой пропускания;
- более эффективная передача данных между системной и видеопамятью;
- отсутствие необходимости в графических API с избыточностью и накладными расходами;
- линейная адресация памяти, возможность записи по произвольным адресам;
- аппаратная поддержка целочисленных и битовых операций.

Основные ограничения CUDA:

- отсутствие поддержки рекурсии для выполняемых функций;
- минимальная ширина блока в 32 потока;
- закрытая архитектура CUDA, принадлежащая NVIDIA.

Основные преимущества CUDA вытекают из того, что эта архитектура спроектирована для эффективного использования неграфических вычислений на GPU и использует язык программирования C, не требуя переноса алгоритмов в удобный для концепции графического конвейера вид. CUDA предлагает способ вычислений на GPU без применения графических API, предлагающий произвольный доступ к памяти (scatter или gather). Такая архитектура использует все исполнительные блоки, а также расширяет возможности за счёт целочисленной математики и операций битового сдвига.

В среднем, при переносе вычислений на GPU, во многих задачах достигается ускорение в 5-30 раз, по сравнению с быстрыми универсальными процессорами. Самые большие цифры достигаются на коде, который не очень хорошо подходит для расчётов при помощи блоков SSE, но вполне удобен для GPU.

Некоторые примеры ускорений синтетического кода на GPU против SSE кода на CPU (по данным NVIDIA):

- флуоресцентная микроскопия: 12x;
- молекулярная динамика (non-bonded force calc): 8–16x;
- электростатика (прямое и многоуровневое суммирование Кулона): 40–120x и 7x.

Для пользователей, не имеющих отношения к программированию, CUDA так же может быть полезна, например, для ускорения рендеринга видео или 3D графики, а также для увеличения частоты обновления изображения в современных компьютерных играх. Все современные программы нелинейного видеомонтажа имеют поддержку CUDA.

**Я.А. Юницкий (УО «ГГУ им. Ф.Скорины», Гомель)**

Науч. рук. **А.В. Воружев**, канд. техн. наук, доцент

## **НАПИСАНИЕ ПРИЛОЖЕНИЯ ИДЕНТИФИКАЦИИ ПРОЦЕССОРА С ПОМОЩЬЮ ИНСТРУКЦИИ CPUID**

CPUID – ассемблерная мнемоника инструкции процессоров x86, используется для получения информации о процессоре. Используя её, программа может определить тип процессора и его возможности. Процессоры поддерживают инструкцию CPUID (как Intel, так и AMD), начиная с пятого поколения (Pentium) и поздних моделей i486.