**Принципы Фон-Неймана в графических процессорах**

1. Принцип массивных параллельных вычислений. GPU изначально создавался, как многоядерная структура, предназначенная для обработки компьютерной графики, поэтому архитектура GPU нацелена на увеличение скорости расчета текстур и сложных графических объектов.
2. Принцип условного перехода. (обеспечение возможности перехода на тот, или иной участок кода)
3. Принцип хранимой программы. (Программа хранится там же где и данные и в принципе от данных ничем не отличается, они не различимы)
4. Принцип использования двоичной системы счисления. Т.е. всё кодируется в двоичной системе счисления.
5. -!!!Принцип программного управления вычислениями. (Последовательное (учитывая параллельные вычисления) выполнение команд в соответствии с заданной программой)
6. Ограниченный набор специализированных команд. Это ускоряет дешифровку и их реализацию.

**Единицы измерения пропускной способности информационных каналов**

Пропускная способность информационных каналов – это максимально возможная скорость передачи информации по каналу.

1. 1 кбит/с = 1 килобит/c = 1024 бит/c

1 мбит/с = 1 мегабит/с = 1048576 бит/с

Единицы измерения скорости передачи информации за секунду.

Для показания скорости может использоваться любая приставка кило- (2^10) мега- (2^20) гига- (2^30) тера- (2^40) и т.д. аналогично системе в физике, вместе со словом бит или байт (= 8 бит), в секунду.

1. characters per second – количество символов, кодируемых двоичным кодом, передаваемых в одну секунду.

Дальше единицы измерения не связаны напрямую с передачей информации.

1. flops - единица используемая для измерения производительности компьютеров, показывающая, сколько операций с плавающей запятой выполняет вычислительная система за 1 секунду. – это своего рода «пропускная способность» процессора, количество информации, которую он может через себя пропустить и обработать.
2. Пропускная способность нейронной сети измеряется 2-мя способами:
	1. среднее арифметическое между объёмами обрабатываемой и создаваемой информации нейронной сетью за единицу времени.
	2. CPS - connections per second – количество соединений в секунду между отдельными элементами нейронной сети.

Здесь также под скоростью можно понимать количество информации, которую принимает и обрабатывает нейронная сеть.

Пропускная способность канала определяется двумя параметрами: частота (сколько раз информацию можно поместить в канал) и количество информации, которое может быть закодировано за один раз.

**Особенности вычислений на ЭВМ арифметических выражений**

1. При вычислении выражений процессору разрешаются эквивалентные преобразования при условии, что соблюдается неприкосновенность скобок и что целые операнды не смешиваются с операндами других типов так, чтобы мог получиться неэквивалентный результат.
2. Возможно переполнение при выполнении целочисленных операций, что приводит с математической точки зрения к взятию остатка от деления на соответствующее число, которое равно 2^(максимальное число разрядов, отводимое на хранение операнда).
3. Результат арифметических операций чисел с плавающей запятой может:
	1. иметь абсолютное значение, больше M (максимального числа) - машинное переполнение;
	2. иметь ненулевое значение, меньшее m (минимального числа) по абсолютной величине - машинный нуль;
4. Операция вычисления выражения с вещественными числами с фиксированной запятой, объединяет недостатки целочисленных операций, для своей целой части и операций с плавающей запятой для своей дробной части.

Таким образом, при вычислении на ЭВМ важно, над чем именно выполняется операция, т.е. максимальное число разрядов, отводимое на операцию, целое число или вещественное с плавающей запятой или вещественное с фиксированной запятой.

Так же после всех операций устанавливается регистр флагов, по которому можно отследить параметры выполнившейся операции, в частности выяснить произошло липереполнение.

**Представление данных в современных вычислениях**

Любые данные в компьютере так или иначе представляются числами, после чего различается лишь трактовка этих чисел. Эти числа могут кодировать целое число, число с плавающей запятой, число с фиксированной запятой, буквы алфавитов различных языков, изображения, музыку, команды... Они так же могут шифровать массивы этих элементарных объектов, матрицы…

Для шифрования каждого вида данных существуют стандарты (хотя, никто не запрещает придумать свой собственный тип кодировки, проблема будет лишь в том, что его никто не будет поддерживать). Так например для кодирования текстовых данных могут использоваться кодировки ACII, UTF-8, …

Графические данные разбиваются на элементы – пиксели, и каждому пикселю присваивается его цвет, способы задания цвета так же могут отличаться, напр. они могут быть в формате RGB – так называемая аддитивная система, а могут быть в формате CMYK – субтрактивная система (Cyan, Magenta, yellow), или модель HSV(HIS) (Hue, Saturation, Value(Intensity) ) (Тон, Насыщенность, Интенсивность). Форматов для хранения изображений существует так же большое количество, напр. bmp, jpeg, …

Звук, являясь аналоговым сигналом, прежде чем сохранить в памяти компьютера, необходимо оцифровать, т.е. снимая с некоторой постоянной частотой показания этого сигнала и записывая значения в данных конкретный момент времени в виде двоичного кода. После этого звук можно преобразовывать, работая с ним как с обычными числами.

Со всеми этими данными можно производить операции. Таким образом, мы имеем объекты – т.е. некоторые данные и множество операций, которые можно с ними производить.

Например, объект число можно складывать, вычитать, умножать, делить, брать от него синус, … (Заметим, что конкретно в данном случае я рассматриваю не только элементарные операции, которые может произвести процессор, но и более сложные, которые реализуются сведением к элементарным директивам ассемблера, например, взятие синуса)

В зависимости от вычислительной системы, операции, которые может выполнять процессор могут отличаться, например в SIMD системе операции для работы с массивами (матрицами) чисел, но не с отдельным числом, в такой системе как правило работают с объектами типа массив или матрица, не рассматривая подробно их внутреннюю структуру, а в SISD системе процессор работает с отдельными числами.

Такой тип данных как команды (для процессора, или какого-нибудь интерпретируемого языка (напр. JavaScript)) – тоже кодируются двоичным кодом, и их тоже в некоторых случаях (когда система этого не запрещает) можно менять, тем самым создавая самомодифицирующуюся программу.

Если же подняться на программный уровень, то можно рассмотреть объектно-ориентированное программирование, и такие языки как С++, где основная идея заключается в создании сложных объектов, в которых описывается, какие данные они содержат, и какие операции над ними можно совершать. Например мы можем создать класс matrix, а в ней создать метод, который будет считать определитель этой матрицы, таким образом над объектом matrix можно произвести операцию – посчитать определитель.