

2. Рудик А. В. Методи оцінки просторового положення об'єктів. Інтегровані інтелектуальні робототехнічні комплекси (ІРТК-2016). Матеріали 9-ої міжнародної НПК. Київ: НАУ, 2016. С. 31-33.
3. Рудик А. В. Багатофункціональні сенсори для мобільної робототехніки. Вісник Інженерної академії України. 2016. №1. С. 30-36.
4. Голован А. А. Математические основы навигационных систем. Ч. 1. Математические модели инерциальной навигации. А. А. Голован, Н. А. Парусников. М.: МГУ, 2011. 136 с.
5. Рудик А. В. Використання медіанної та діагностичної фільтрацій в мобільних робототехнічних комплексах для попередньої обробки сигналів. А. В. Рудик. Збірник наукових праць Одеської державної академії технічного регулювання та якості. 2016. № 1 (8). С. 73-78.
6. Рудик А. В. Синтез та моделювання цифрових фільтрів програмними засобами MATLAB. А. В. Рудик. Вимірювальна та обчислювальна техніка в технологічних процесах (Хмельницький). 2017. № 3. С. 87-93.

РЕАЛІЗАЦІЯ ПАРАЛЕЛІЗМУ ПОТОКІВ КОМАНД І ДАНИХ ГРАФІЧНИХ ПРОЦЕСОРІВ

Завальнюк Є. К.

*здобувач третього (освітньо-наукового) рівня вищої освіти
Вінницького національного технічного університету*

Романюк О. Н.

*доктор технічних наук, професор,
завідувач кафедри програмного забезпечення,
Вінницького національного технічного університету*

Вступ. Рендеринг [1, с. 63] є етапом графічного конвеєра, що включає застосування багатьох складних обчислень, зокрема, матричних і векторних операцій. Матричні та векторні операції рендерингу застосовуються для переходу між системами координат, здійснення проєкцій, трансформацій об'єктів, розрахунку моделей відбивної здатності поверхні. Важливою є можливість їх оптимального обчислення. Тому для реалізації трудомістких графічних операцій використовуються спеціальні архітектури графічних процесорів [2] (GPU). Архітектури GPU можуть бути класифіковані за принципом роботи складових процесорів згідно з таксономією Флінна [3].

Мета. Проаналізувати особливості таксономії Флінна та визначити найбільш доцільну організацію роботи процесорів GPU.

Основна частина. Згідно з Майклом Флінном [3] комп'ютерні системи поділяються залежно від кількості одночасно використаних потоків даних та інструкцій.

SISD [3] (single instruction stream, single data stream) – на одному процесорі одна інструкція виконується над одним потоком даних. Є найпростішим типом архітектури комп'ютерної системи, однак є найменш потужним. Типовим прикладом SISD-архітектури є перші одноядерні персональні комп'ютери.

SIMD [3] (single instruction stream, multiple data stream) – на багатьох обчислювальних блоках (ОБ) та ж інструкція виконується над різними потоками даних. Архітектура є доцільною для завдань, що потребують багато векторних і матричних обчислень. Наприклад, SIMD використовується при обробці сигналів і зображень.

Підвидом SIMD є SIMT [4] (single instruction, multiple threads – «одна інструкція, багато потоків»). При використанні SIMT кожен паралельний обчислювальний блок має свою пам'ять і файл реєстру, використовується багатопотоковість.

MISD [3] (multiple instruction stream, single data stream) – на багатьох ОБ різні інструкції виконуються над одним потоком даних. Наприклад, виконується набір математичних операцій над тим же набором даних, здійснюється контроль помилок над даними. Архітектура рідко використовується.

MIMD [3] (multiple instruction stream, multiple data stream) – на автономних ОБ багато інструкцій виконуються над багатьма потоками даних. Архітектура характерна для паралельних, розподілених і високопродуктивних обчислень.

Сучасні GPU Nvidia складаються з великої кількості поточкових мультипроцесорів [4] (багатоядерних процесорів). Наприклад, архітектура GPU Ada Lovelace [5] вміщує 12 кластерів графічної обробки з 12-а поточковими мультипроцесорами.

Потокові мультипроцесори сучасних GPU зазвичай використовують архітектуру SIMT [4]. Написана для GPU програма ділиться на блоки потоків. Кожен блок потоків призначається для обробки на певному поточковому мультипроцесорі GPU. Блоки потоків поділяються на «варпи» [4] по 32 потоки. У межах «варпу» одна інструкція багатопотоково виконується над різними даними (SIMT). Формується черга із «варпів» для обробки на поточковому мультипроцесорі. Якщо інструкція «варпа» чекає отримання даних, то запускається інструкція іншого «варпа». Перемикання між «варпами» забезпечують ефективне використання обчислювальних ресурсів.

Даний підхід є високопродуктивним для обчислення матричних і векторних графічних операцій.

Висновок. Використання у мультипроцесорах GPU парадигми SIMT забезпечує високопродуктивну візуалізацію тривимірних сцен за рахунок застосування багатопотокової обробки спільних інструкцій.

ЛІТЕРАТУРА

1. Романюк О. Н. Комп'ютерна графіка. Навчальний посібник. Вінниця: ВДТУ, 1999. 130 с.
2. Завальнюк Є. К., Романюк О. Н., Снігур А. В., Шевчук Р. П. Аналіз сучасних архітектур GPU. Стан, досягнення та перспективи інформаційних систем і технологій. Одеса: ОНТУ, 2023. С. 302 – 303.
3. Computer Architecture | Flynn's taxonomy. GeeksForGeeks: web site. URL: <https://www.geeksforgeeks.org/computer-architecture-flynns-taxonomy/> (accessed on: 28.08.2023).
4. Yousefzadeh-Asl-Miandoab E. How Do GPUs Work? Medium: web site. URL: <https://medium.com/mlearning-ai/how-do-gpus-work-13bb243c17d> (accessed on: 28.08.2023).
5. NVIDIA ADA GPU ARCHITECTURE. Nvidia: web site. URL: <https://images.nvidia.com/aem-dam/Solutions/Data-Center/14/nvidia-ada-gpu-architecture-whitepaper-v2.1.pdf> (accessed on: 28.08.2023).