

## ОРГАНІЗАЦІЯ ПІДСИСТЕМИ ТЕКСТУРУВАННЯ ГРАФІЧНИХ ПРОЦЕСОРІВ

Вінницький національний технічний університет

### *Анотація.*

*Розглянуто архітектурні особливості апаратних блоків текстуровання в сучасних графічних процесорах. Проаналізовано структуру Texture Mapping Units, принципи роботи блоків адресації та фільтрації, а також ієрархію кеш-пам'яті, що забезпечує ефективну вибірку текстурних даних. Описано взаємодію текстурних модулів із шейдерними ядрами та підсистемою пам'яті в межах графічного конвеєра. Наведено особливості реалізації текстурних підсистем у сучасних архітектурах GPU та визначено їх вплив на продуктивність рендерингу.*

Ключові слова: графічний процесор; GPU; текстуровання; Texture Mapping Unit; кеш-пам'ять; графічний конвеєр; шейдерні ядра; фільтрація текстур; адресація текстур; архітектура GPU.

### *Abstract*

*The architectural features of hardware texturing units in modern graphics processors are considered. The structure of Texture Mapping Units, the principles of operation of addressing and filtering units, as well as the cache memory hierarchy that ensures efficient sampling of texture data are analyzed. The interaction of texture modules with shader cores and the memory subsystem within the graphics pipeline is described. The features of the implementation of texture subsystems in modern GPU architectures are presented and their impact on rendering performance is determined.*

**Keywords** graphics processor; GPU; texturing; Texture Mapping Unit; cache memory; graphics pipeline; shader cores; texture filtering; texture addressing; GPU architecture.

### **Вступ**

Сучасні графічні процесори є високопродуктивними паралельними обчислювальними системами, орієнтованими на обробку великих потоків графічних даних у режимі реального часу. Однією з ключових підсистем GPU є апаратні блоки текстуровання, що забезпечують вибірку, адресацію та фільтрацію текстур при формуванні зображення [1]. Дослідження їх архітектурної організації є важливим для розуміння принципів підвищення продуктивності та енергоефективності сучасних графічних систем.

### **Результати дослідження**

У графічних процесорах (GPU) апаратні блоки, що відповідають за текстуровання, являють собою центральний елемент реалізації тривимірного рендерингу, оскільки вони забезпечують вибірку, трансформацію та фільтрацію текстурних даних без залучення центрального процесора. В контексті архітектури GPU блоки мапінгу текстур (TMU) виконують основну обробку растрових зображень, перетворюючи їх у значення, які накладаються на полігони тривимірної сцени під час рендерингу. За визначенням, TMU відповідають за обчислювальні операції з обробки текстур, включно з обчисленням координат та базовими фільтраційними операціями, що дозволяє графічному процесору ефективно «наклеювати» текстуру на задану поверхню [1]. Такі блоки є окремими логічними одиницями всередині конвеєра GPU, що працюють у тісній взаємодії зі шейдерними ядрами та іншими обчислювальними компонентами.

Однією з апаратних підсистем є кеш текстур, що представляє собою спеціалізовану ієрархічну пам'ять, оптимізовану під доступ до двовимірних та тривимірних масивів даних. Відомо, що кеш текстур GPU поділяється на локальний L1 кеш, прив'язаний до кожного TMU або мультипроцесору, та більш високорівневий спільний кеш L2, який обслуговує запити кількох блоків одночасно, що

сприяє суттєвому зниженню затримок доступу до глобальної пам'яті та підвищує загальну пропускну здатність при виконанні паралельних доступів до текстур [2]. Ефективність такого кешування є предметом досліджень у контексті оптимізації доступу пам'яті, оскільки нерегулярні шаблони доступу можуть значно знизити частоту попадання в кеш і, відповідно, продуктивність [3].

У сучасних GPU TMU не лише здійснюють доступ до даних, але й виконують різні режими адресації та фільтрації текстур. Адресаційні блоки дозволяють обробляти координати текстур так, щоб підтримувати режими wrap, clamp та mirror, що є необхідним для правильного накладання текстур у різних умовах рендерингу. Також апаратно реалізовані фільтраційні блоки виконують інтерполяцію вибірки texel для таких режимів, як білінійна та анізотропна фільтрація, що покращує якість текстур при віддаленні або під кутом щодо камери сцени. Значення цих апаратних механізмів важко переоцінити, оскільки їх відсутність або програмна реалізація значно знизили б продуктивність систем рендерингу в реальному часі [1].

Технічні дослідження показують, що продуктивність апаратних блоків текстуровання GPU істотно залежить від кількості текстурних блоків і параметрів їх взаємодії із шейдерними ядрами. Наприклад, збільшення числа TMU в архітектурі мобільних GPU корелює з підвищенням продуктивності при виконанні графічних задач, що демонструють важливість цих апаратних одиниць при побудові моделей продуктивності GPU для рендерингу. Деякі моделі навіть використовують параметри TMU як основу для прогнозування частоти кадрів у графічних застосунках, що підкреслює їхню критичну роль у загальній архітектурі системи [4].

Апаратна організація кешів та їх політики також є важливими для оптимізації продуктивності текстуровання. Зокрема, дослідження у сфері оптимального кешування текстур пропонують нові підходи до обробки кешу L2 для зменшення трафіку пам'яті та енергоспоживання, що є надзвичайно актуальним для мобільних GPU та вбудованих систем, де енергоефективність є важливою вимогою. Це підкреслює, що апаратні блоки для текстуровання не тільки прискорюють виконання графічних задач, але й мають прямий вплив на енергоспоживання та продуктивність систем у цілому [5].

На додаток до згаданих елементів, в конвеєрі GPU важлива координація між TMU, кешами та іншими апаратними блоками, такими як блоки растеризації та шейдерні ядра. Це забезпечує паралельну обробку текстур, де вибірка, фільтрація та обробка даних відбуваються одночасно для великої кількості пікселів. Такий рівень апаратної інтеграції дозволяє досягти високих значень пропускну здатності та продуктивності при рендерингу складних сцени у реальному часі. Саме така архітектура лежить в основі сучасних GPU як у десктопних системах, так і у мобільних або вбудованих платформах, що зумовлює їх широке застосування в графіці, візуалізації та обчисленнях загального призначення [3, 5].

GPU містить апаратні блоки, що відповідають за текстуровання, які забезпечують вибірку, трансформацію та фільтрацію текстурних даних без участі центрального процесора [4]. Текстуровання є ключовим елементом графічного конвеєра, оскільки визначає візуальну якість сцени та деталізацію поверхонь об'єктів. Блоки мапінгу текстур (texture mapping units, TMU) обробляють растрові зображення, перетворюючи їх у значення для накладання на полігони сцени. Кожен TMU отримує текстурні координати від шейдерного ядра, обчислює відповідні texel і передає результати в блоки фільтрації. Сучасні GPU містять десятки TMU, що дозволяє обробляти сотні або тисячі пікселів одночасно. Координати текстури обробляються блоками адресації, які трансформують нормалізовані координати в індекси texel і забезпечують режими wrap, clamp та mirror для коректного накладання текстур. Кеш текстур (texture cache) має два рівні: L1 локальний для TMU та L2 спільний для декількох блоків. Локальні кеші забезпечують швидкий доступ до часто використовуваних texel, а L2 кеш оптимізує запити до великих текстур [2, 3]. Кешування підвищує ефективність текстуровання та знижує затримки доступу до глобальної пам'яті. Після вибірки texel TMU передають дані в блоки фільтрації, які виконують білінійну, трилінійну та анізотропну інтерполяцію для плавного переходу між texel і підвищення якості відображення текстур. Для великих текстур GPU використовує ієрархічну пам'ять mipmaps, tiled memory layout та апаратне стиснення текстур для зменшення обсягу даних і підвищення пропускну здатності. Важливою особливістю є інтеграція TMU, кешів, шейдерних ядер і блоків растеризації, що забезпечує паралельну обробку великої кількості пікселів. Для оптимізації продуктивності застосовуються техніки попереднього завантаження текстур (prefetching), групування запитів (batching) та раннє відсіювання невидимих пікселів (early culling). У архітектурах NVIDIA Turing та Ampere кожен Streaming Multiprocessor (SM) містить численні TMU з кешами L1 та L2 і підтримкою апаратної фільтрації та стиснення текстур. В

архітектурах AMD RDNA та RDNA2 кожен Compute Unit (CU) має TMU з локальним L1 кешем та спільним L2 кешем, підтримкою mipmaps, апаратної фільтрації та Variable Rate Shading для оптимізації продуктивності. Конвеєр текстуровування включає шейдерні ядра, блок адресації, L1 кеш, L2 кеш, TMU/Texel Fetch Unit, блок фільтрації, фреймбуфер і оптимізаційні блоки Prefetching, Early Culling та Watching, що взаємодіють із TMU та кешами для підвищення продуктивності та якості рендерингу сцени. Оптимізація поєднання апаратних блоків текстуровування визначає швидкість рендерингу, якість відображення матеріалів і ефективність роботи GPU в цілому.

### Висновки

Апаратні блоки текстуровування є важливим компонентом сучасних графічних процесорів, що забезпечує високошвидкісну вибірку та обробку текстурних даних у режимі реального часу. Їх структурна організація, кількість Texture Mapping Units та ефективність кеш-підсистеми безпосередньо впливають на продуктивність і якість рендерингу. Інтеграція блоків адресації, фільтрації та кешування в межах єдиного графічного конвеєра дозволяє досягти значного рівня паралелізму обробки. Подальший розвиток архітектур GPU спрямований на підвищення пропускної здатності текстурних підсистем і зниження затримок доступу до пам'яті.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Texture mapping unit // Wikipedia : вебсайт. URL: [https://en.wikipedia.org/wiki/Texture\\_mapping\\_unit](https://en.wikipedia.org/wiki/Texture_mapping_unit) (дата звернення: 01.02.2026).
2. Govindaraju N., Larsen S., Gray J., Manocha D. Memory model for scientific algorithms on GPUs // ResearchGate. 2006. URL: <https://www.researchgate.net/figure/Texture-caches-on-a-commodity-GPU-NVIDIA-GeForce-7800> (дата звернення: 01.02.2026).
3. Wang J., Yang F., Cao Y. A cache-friendly sampling strategy for texture-based volume rendering on GPU // Visual Informatics. 2017. DOI: 10.1016/j.visinf.2017.08.001.
4. Yun J., Lee J., Kim C. G. et al. A Practically Applicable Performance Prediction Model Based on TMUs for Mobile GPUs // IEEE Access. 2019.
5. Joseph D., Aragon J. L., Parcerisa J.-M., Gonzalez A. Quasi-Optimal L2 Caching for Textures in GPUs. Universitat Politecnica de Catalunya, 2023. URL: [https://webs.um.es/jlaragon/papers/joseph\\_PACT23.pdf](https://webs.um.es/jlaragon/papers/joseph_PACT23.pdf) (дата звернення: 01.02.2026).

**Романюк Олександр Никифорович** - д-р техн. наук, професор, завідувач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: rom8591@gmail.com.

**Майданюк Володимир Павлович** - канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: maidaniuk2000@gmail.com.

**Новосельцев Олександр Олександрович** - аспірант кафедри програмного забезпечення інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: sasha\_novoseltsev@icloud.com.

**Romanyuk Oleksandr Nikiforovich** - Dr. of Technical Sciences, Professor, Head of the Department of Software, Vinnytsia National Technical University, Vinnytsia, e-mail: rom8591@gmail.com.

**Maidaniuk Volodymyr Pavlovych** - Candidate of Technical Sciences, Associate Professor of the Department of Software, Vinnytsia National Technical University, Vinnytsia, e-mail: maidaniuk2000@gmail.com.

**Novoseltsev Oleksandr Oleksandrovich** - Postgraduate Student of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: sasha\_novoseltsev@icloud.com.