

УДК 616.2

DOI <https://doi.org/10.32782/2663-5941/2024.6.1/24>**Саражинський В.О.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**Лебедев Д.Ю.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**Антонюк О.І.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

РЕАЛІЗАЦІЯ АЛГОРИТМУ ВЕРИФІКАЦІЇ, ЯКИЙ ҐРУНТУЄТЬСЯ НА РАНДОМІЗАЦІЇ З ОБМЕЖЕННЯМИ, ДЛЯ ПЕРЕВІРКИ БЛОКУ DMA СИНТЕЗОВАНОГО МОВАМИ ОПИСУ АПАРАТУРИ

Стаття присвячена розробці та реалізації алгоритму верифікації блоку Direct Memory Access (DMA), що є критично важливим компонентом у сучасних цифрових системах. У статті розкрито значення верифікації як одного з ключових етапів у розробці електронних систем для забезпечення їхньої надійності, відповідності технічним вимогам і уникнення помилок на етапах виробництва. Особливу увагу приділено DMA-контролерам, які відповідають за передачу даних між пам'яттю та периферійними пристроями без участі центрального процесора. Некоректна робота таких компонентів може спричинити втрати даних, зниження продуктивності та нестабільність системи, що обумовлює необхідність ефективної верифікації.

В ході статті визначено мету дослідження – реалізацію алгоритму верифікації блоку DMA на основі рандомізації з обмеженнями для перевірки функціональності та відповідності специфікаціям. З'ясовано, що ключовими етапами процесу верифікації є створення верифікаційного плану, розробка тестового середовища, створення тестів, проведення симуляцій, збір функціонального покриття та аналіз результатів. У роботі також розкрито методику побудови верифікаційного середовища, яке використовує компоненти Universal Verification Methodology (UVM) для забезпечення максимальної гнучкості та ефективності тестування.

Розкрито основні аспекти реалізації алгоритму верифікації, включаючи створення тестових послідовностей, інтеграцію верифікаційних модулів, а також аналіз результатів функціонального покриття. Верифікація була проведена для модуля АНВ DMA Controller з відкритим кодом, що використовує АНВ-Lite інтерфейси. У результаті досягнуто функціонального покриття на рівні 95%, що підтверджує ефективність запропонованого алгоритму та верифікаційного середовища.

З'ясовано, що однією з причин недосягнення 100% покриття є ймовірні баги в дизайні, що потребують подальшого вдосконалення. Визначено перспективи розвитку, які включають оптимізацію алгоритму рандомізації, розширення тестових сценаріїв для досягнення 100% покриття, а також інтеграцію нових перевірок для більш детального тестування.

Запропонований метод демонструє високу гнучкість, що дозволяє адаптувати його до різних умов тестування та застосовувати для верифікації інших компонентів цифрових систем. Результати дослідження можуть бути корисними для інженерів у сфері розробки апаратного забезпечення, забезпечуючи стабільну роботу сучасних цифрових систем.

Ключові слова: верифікація, DMA, рандомізація з обмеженнями, функціональне покриття, тестове середовище.

Постановка проблеми. Верифікація цифрових мікросхем є ключовим етапом у розробці сучасних електронних систем, оскільки забезпечує відповідність дизайну технічним вимогам та запобігає появі помилок на пізніх стадіях вироб-

ництва. У випадку DMA (Direct Memory Access) контролерів, які відповідають за передачу даних між пам'яттю та периферійними пристроями без втручання центрального процесора, забезпечення їхньої коректної роботи є критично важливим

завданням. DMA блокується помилками дизайну, що може призводити до некоректної роботи системи чи виникнення часових збоїв. Некоректна робота DMA може призвести до серйозних проблем в системі, таких як втрати даних, зниження продуктивності та нестабільність роботи. Тому розробка ефективних алгоритмів верифікації блоку DMA є важливою задачею для забезпечення надійності та стабільності цифрових систем.

Аналіз останніх досліджень і публікацій.

В даному розділі описано загальний алгоритм верифікації цифрових мікросхем [2]. Даний алгоритм може змінюватись та доповнюватись в залежності від складності мікросхеми. Блок-схему алгоритму наведено на рисунку 1.

1. Створення верифікаційного плану

Першим кроком у процесі верифікації є підготовка плану верифікації, який тісно пов'язаний зі специфікацією проекту.

Верифікаційний план включає тест-план (список тестових сценаріїв, спрямованих на особливості дизайну), планування функціонального покриття, призначення модулів/блоків інженерам з верифікації. План верифікації також передбачає планування повторного використання компонентів верифікації під час перевірки на рівні системи.

2. Розробка тестового середовища

У рамках розробки тестового середовища інженери з верифікації розробляють компоненти верифікаційного середовища, інтерфейси для з'єднання тестового середовища з дизайном, інтегрують готові верифікаційні блоки (VIPs) в загальне середовище, розробляють модулі контролю (checkers) тощо.

3. Створення тестів

На даному етапі створюються рандомізовані або спеціальні тести написані для однієї чи кількох функцій у дизайні у відповідності до тест-плану. Тести запускають послідовності на основі UVM для симуляції необхідних сценаріїв.

4. Симуляція

На цьому етапі інженери перевіряють, чи верифікована певна функція чи ні. Якщо ні, знову змінюються тестові послідовності та ще раз запускається симуляція. За допомогою модулю сповіщення (scoreboard) повідомляється про помилку, якщо бажаний дизайн не працює належним чином. В результаті аналізу помилок та часових діаграм сигналів оцінюється середовище верифікації та повідомляється про помилку команді проектувальників, якщо виявилось, що це проблема дизайну, інакше моделювання запускається повторно після виправлення помилок верифікаційного середовища.

5. Збір покриття

Наступним важливим кроком є збір функціонального покриття. Це вирішальний етап, оскільки на цьому етапі вирішується чи достатньо верифіковано функції мікросхеми (досягнуто необхідного відсотку покриття).

6. Аналіз покриття та доповнення тестів

Аналізуючи покриття нерідко виникає необхідність додавання спеціальних тестів або запуску багатьох тестів регресивно.

Процес верифікації завершується при досягненні 100% покриття, або якщо це неможливо чи займатиме забагато часу – при досягненні відсотку покриття не меншого ніж зазначено в специфікації.

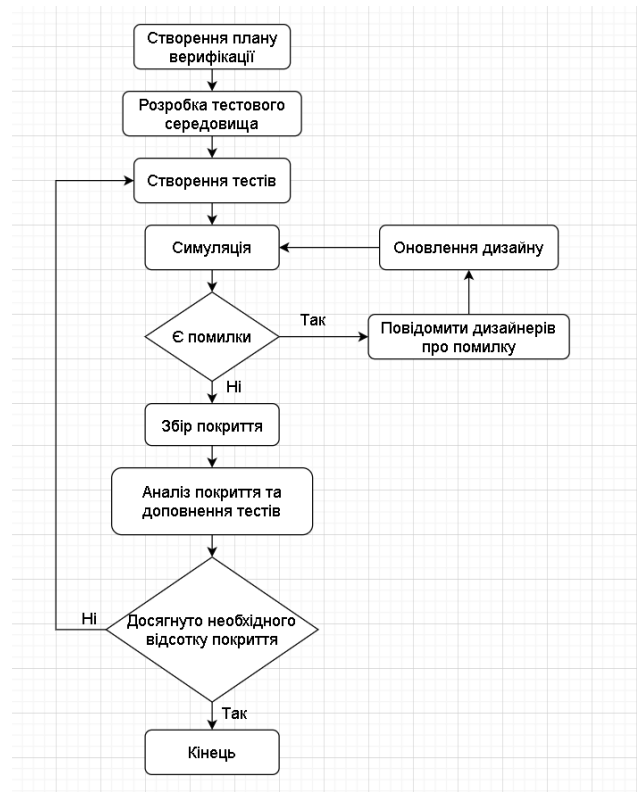


Рис. 1. Блок-схема алгоритму верифікації

Постановка завдання. Метою даної роботи є реалізація алгоритму верифікації блоку DMA, заснованого на рандомізації з обмеженнями, для перевірки його функціональності та відповідності специфікаціям. Основні завдання включають:

1. Реалізація алгоритму верифікації, що використовує рандомізацію з обмеженнями.
2. Створення верифікаційного середовища.
3. Проведення верифікації блоку DMA та аналіз досягнутого функціонального покриття.

Виклад основного матеріалу. У якості DMA блоку використовується модуль АНВ DMA Controller [3] з відкритим кодом написаним мовою Verilog. Цей модуль створений на основі АНВ-Lite

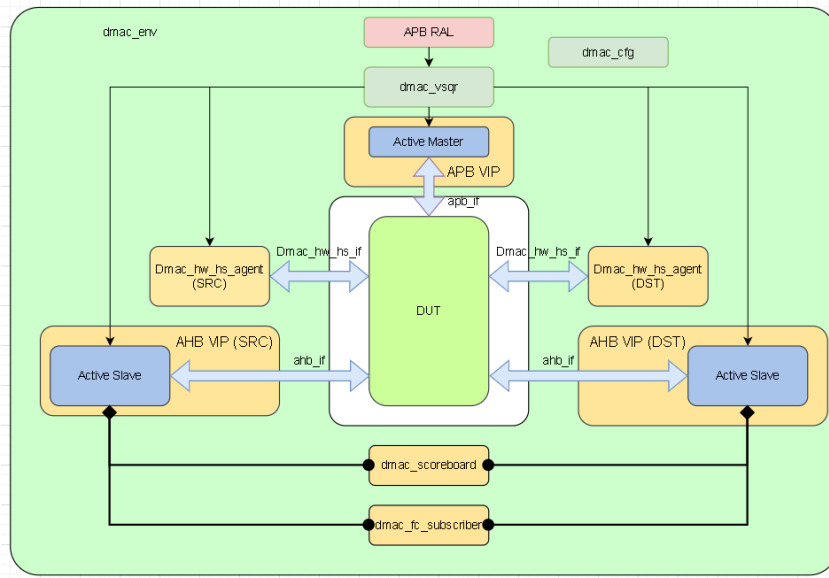


Рис. 2. UVM середовище тестування DMA

інтерфейсів та має опцію HW handshaking що вділяє його з-поміж інших варіантів які є у відкритому доступі.

Для побудови тестового стенду необхідні наступні компоненти:

1. Симулятор, який підтримує мову SystemVerilog версії IEEE-1800-2017.

2. Компілятор який підтримує бібліотеку UVM 1800.2-2020 – IEEE, та з підтримкою рандомізації та оцінкою покриття (coverage).

3. Скрипти для створення команд для компіляції та симуляції середовища.

У даній роботі буде використовуватися Synopsys VCS у якості симулятора, та Synopsys Verdi для перегляду сигналів (waveforms) та результатів покриття, оскільки дані інструменти задовольняють вищеперерахованим вимогам.

Процес створення верифікаційного середовища (рис. 2) включає наступні етапи:

1. Ініціалізація середовища: На даному етапі створюється саме середовище (dmac_env), файл конфігурації (dmac_cfg) та інтегруються VIPs. В даному прикладі використовуються два АНВ Slave VIP від Synopsys для симуляції передачі даних від одного до іншого та один APB Master VIP від Synopsys для конфігурації DMA. Також інтегровано два агенти для HW handshaking.

2. Розробка пошукових алгоритмів (sequence): На даному етапі створюються випадкові та цільові тестові послідовності, які зможуть сконфігурувати DMA контролер певним чином та просимулювати той чи інший випадок.

3. Налаштування модулю сповіщення (scoreboard): Один з найскладніших етапів розробки верифікаційного середовища – це продумування та створення тестових модулів (checkers). В даному середовищі більшість тестових модулів перевіряють коректність роботи DMA, порівнюючи дані, які були записані в регістри DMA контролера під час конфігурації з реальними даними, отриманими з DUT.

4. Запуск тестів: Для запуску пошукових алгоритмів (sequence) створюються тести. Для більшого покриття зазвичай створюють скрипти які регресивно запускають необхідні тести в необхідній кількості.

5. Збір покриття та доповнення тестів: Планування та збір функціонального покриття це найвідповідальніший етап верифікації, оскільки саме на цьому етапі визначається наскільки верифікований дизайн, чи всі його функції перевірені та як саме.

Верифікація DMA-блоку була проведена за допомогою розробленого алгоритму. Функціональне покриття досягло 95%, що свідчить про високу ефективність верифікаційного середовища. Результати функціонального покриття зображено на рисунку 3.

Незважаючи на досягнутий результат, покриття не досягло 100% ймовірно через баги, які ще залишились в дизайні.

Висновки. У результаті проведеного дослідження реалізовано алгоритм верифікації блоку DMA, заснований на рандомізації з обмеженнями. Запропонований метод дозволяє ефективно перевіряти функціональність DMA-блоку,

Avg. Group Score:97.86% U+C:45 U:3 C:42 X:0					
Avg. Group Inst. Score:97.86% U+C:45 U:3 C:42 X:0					
Group	Score	Instances	U+C	U	C
Cg dmac_pkg::dmac_fc_subscriber::dmac_hready_ctrl_cg	100.00%		6	0	
Cr dst_hready_num_wait_cycles	100.00%		3	0	
Cr src_hready_num_wait_cycles	100.00%		3	0	
Cg dmac_pkg::dmac_fc_subscriber::dmac_transaction_cg	95.71%		39	3	
Cr block_ts	57.14%		7	3	
Cr dar	100.00%		5	0	
Cr dst_addr_inc	100.00%		1	0	
Cr dst_tr_width	100.00%		3	0	
Cr fifo_th	100.00%		4	0	
Cr sar	100.00%		5	0	
Cr src_addr_inc	100.00%		1	0	
Cr src_dst_type	100.00%		1	0	
Cr src_tr_width	100.00%		3	0	
Cr mem_per_type_width	100.00%		9	0	

Рис. 3. Результати функціонального покриття

забезпечуючи високе функціональне покриття та виявляючи потенційні помилки в дизайні. Обраний підхід демонструє високу гнучкість та здатність адаптуватися до різних умов тестування, що робить його оптимальним для верифікації DMA-блоків у сучасних цифрових системах.

Для подальшого вдосконалення методики верифікації планується:

1. Збільшення функціонального покриття: Розширення тестових сценаріїв для досягнення 100% покриття.

2. Оптимізація алгоритму рандомізації: Вдосконалення обмежень для більш ефективного генерування тестових послідовностей.

3. Інтеграція додаткових перевірок: Додавання нових тестових модулів для перевірки додаткових аспектів функціонування DMA-блоку.

Таким чином, реалізація запропонованого алгоритму верифікації є важливим кроком у напрямку підвищення надійності та якості цифрових систем, забезпечуючи їхню стабільну та ефективну роботу.

Список літератури:

1. What is functional verification. [Електронний ресурс]. Режим доступу: https://www.cadence.com/en_US/home/explore/functional-verification.html
2. ASIC verification flow. [Електронний ресурс]. Режим доступу: <https://vlsiverify.com/asic-flows/asic-verification-flow/>
3. Opencores.org project – DMA AHB – 32 bit build. [Електронний ресурс]. Режим доступу: https://github.com/freecores/dma_ahb/tree/master/src/dma_ahb32

Sarzhynskiy V.A., Lebedev D.Yu., Antonyuk A.I. IMPLEMENTATION OF A CONSTRAINED RANDOM ALGORITHM FOR VERIFICATION OF THE DMA CONTROLLER MODULE SYNTHESISED IN HARDWARE DESCRIPTION LANGUAGES

The article is devoted to the development and implementation of a verification algorithm for the Direct Memory Access (DMA) block, which is a critical component in modern digital systems. The article reveals the importance of verification as one of the key stages in the development of electronic systems to ensure their reliability, compliance with technical requirements and avoidance of errors at the production stage. Particular attention is paid to DMA controllers, which are responsible for transferring data between memory and peripheral devices without the participation of the central processor. Incorrect operation of such components can lead to data loss, performance degradation, and system instability, which necessitates effective verification.

The article defines the purpose of the study – to implement a randomization-based DMA block verification algorithm with constraints to check functionality and compliance with specifications. It has been found that the key stages of the verification process are the creation of a verification plan, development of a test environment, creation of tests, conducting simulations, collecting functional coverage and analyzing the results. The

paper also reveals a methodology for building a verification environment that uses Universal Verification Methodology (UVM) components to ensure maximum flexibility and efficiency of testing.

The main aspects of implementing the verification algorithm, including creating test sequences, integrating verification modules, and analyzing the results of functional coverage, are revealed. The verification was carried out for the AHB DMA Controller module, an open source module that uses AHB-Lite interfaces. As a result, functional coverage of 95% was achieved, which confirms the effectiveness of the proposed algorithm and verification environment.

It is found that one of the reasons for not achieving 100% coverage is possible bugs in the design that require further improvement. The prospects for development are identified, including optimization of the randomization algorithm, expansion of test scenarios to achieve 100% coverage, and integration of new checks for more detailed testing.

The proposed method demonstrates high flexibility, which allows it to be adapted to different testing conditions and applied to verify other components of digital systems. The results of the study can be useful for engineers in the field of hardware development, ensuring the stable operation of modern digital systems.

Key words: *verification, DMA, random constrained verification, functional coverage, test environment.*