

ОГЛЯД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИКОНАННЯ ПРОЕКТІВ НА ПЛІС

студент Беззабарний Д.І., канд. техн. наук, доц. Воргуль О.В.

Харківський національний університет радіоелектроніки,
кафедра мікропроцесорних технологій і систем, м. Харків, Україна
e-mail: dmytro.bezzabarnyi@nure.ua

Abstract. The purpose of this work is to provide an open source overview of the FPGA software which can be found on the Internet. Some comparison by complexity of operation, possibilities and usability is performed.

Ключові слова: ПЛІС, програмне забезпечення, проет.

Вступ. Щоб навчитися складати моделі електронних цифрових систем мовою HDL, необхідна практика. Під час навчання мови програмування необхідний базовий інструмент програміста. Таким же чином, для того, щоб вивчити мову HDL, майбутньому спеціалісту потрібен інструмент розробника. Потрібно отримувати практику, а політика фірм розробників, наприклад, Xilinx згодом краще не стає. І софт у них усім добрий, але важкий, і точно не компактний [1-4].

Зі звичайним програмуванням легше. У самому спартанському випадку програмування можна вивчати, маючи текстовий редактор, компілятор, можливо, відладчик, і процесор. Звичайно ж, зараз усі звикли до середовищ розробника, щоб з описом команд, щоб редактор із підсвічуванням синтаксису та вбудованою допомогою. І так, це зручно.

Основна частина. А як для вивчення мови HDL? Роль процесора у цій ситуації може зіграти або симулятор, або живий кристал ПЛІМ чи ПЛІС. Особливістю ПЛІС є те, що результатом є модель електронного приладу, який не є самодостатнім. Для його повноцінної перевірки він повинен бути вбудований у схему або тестовий стенд [5-8].

Адже з програмуванням як. Сидиш за комп'ютером та пишеш програму. І відразу її запускаєш і перевіряєш. А якщо ми створюємо модель електричного приладу? Який мінімальний набір програмного забезпечення потрібен?

З 80-х років минулого століття, часом – бурхливо, часом – неквапливо, розвиток цієї галузі відбувався у різних напрямках [1]. Спочатку пропонувалося в ході створення моделі паралельно створювати тестову програму - тест бенч. Виходить, що модель перевіряється моделлю. У світі HDL інструментом може бути симулятор.

Симулятори створювалися як розробниками апаратного забезпечення – великими фірмами, і фірмами чи співтовариствами, які займаються випуском власних ПЛІМ і ПЛІС.

Що робить симулятор для мови HDL? Arja, ChatGPT з Opera [2] інформує:

Симулятор HDL (мова опису обладнання) – це програмний інструмент, який використовується при цифровому проектуванні та перевірці для моделювання та тестування цифрових схем та систем, що описуються з використанням HDL, таких як VHDL (мова опису апаратного забезпечення надшвидкісних інтегральних схем) або Verilog . Основна мета симулятора HDL – дати розробникам можливість перевірити правильність та функціональність своїх цифрових проектів до їх реалізації на апаратному рівні.

Ось що може симулятор HDL:

1. Поведінкове моделювання. Симулятор HDL може моделювати поведінку цифрової схеми чи системи більш високому рівні абстракції. Це дозволяє розробникам описувати функціональність схеми за допомогою конструкцій HDL та моделювати її поведінку з часом. Це дозволяє їм тестувати та налагоджувати проект, перш ніж переходити до реалізації нижчого рівня.

2. Часовий аналіз та аналіз продуктивності. Симулятори HDL можуть аналізувати часову поведінку проекту, щоб переконатися, що він відповідає бажаним вимогам щодо продуктивності. Вони можуть виконувати аналіз критичного шляху та виявляти потенційні проблеми синхронізації, такі як гонки чи надмірні затримки, які можуть вплинути на загальну функціональність схеми.

3. Генерація тест бенча (тестового стенду). Це HDL модель, що є набором стимулів, що використовуються для управління вхідними даними цифрового проекту під час моделювання. Симулятори HDL надають інструменти автоматичного або ручного створення тестових стендів. Ці тестові стенди дозволяють розробникам перевіряти поведінку своїх проектів у різних вхідних умовах та тестувати різні сценарії, щоб гарантувати правильну функціональність.

4. Налагодження та перегляд сигналів: симулятори HDL надають можливості налагодження, дозволяючи розробникам відстежувати та аналізувати поведінку своїх проектів. Вони можуть встановлювати точки зупинки, покроково виконувати моделювання та перевіряти значення сигналів та змінних у різні моменти часу. Засоби перегляду сигналів – поширена функція HDL-симуляторів, що дозволяє розробникам візуалізувати сигнали та їх переходи з часом.

5. Перевірка та аналіз покриття. Симулятори HDL підтримують методологію перевірки, що дозволяє гарантувати, що проект відповідає зазначеним вимогам. Вони можуть виконати аналіз функціонального покриття, щоб визначити повноту процесу перевірки та виявити будь-які неперевірені частини проекту.

Загалом, симулятор HDL є найважливішим інструментом у процесі цифрового проектування та перевірки, що дозволяє розробникам моделювати, тестувати, налагоджувати та перевіряти поведінку та

продуктивність своїх цифрових схем та систем, описаних з використанням HDL.

Xilinx пропонує Vitis, Vivado та ISE. Ці версії мають вбудовані різні симулятори. ISE легше, простіше, але на нових версіях операційних систем працює лише на віртуальній машині з підтримкою Windows 7. Xilinx вміло керує популярністю більш свіжих продуктів, встановивши більш потужні стимулятори в Vivado та Vitis.

Altera, яка тепер зветься Intel, пропонувала Altera Max Plus та Quartus II. На сайті Intel пропонується Quartus Prime, підтримка попередніх вище згаданих програм припинена.

Крім пари грандів, інтерес становлять ще два розробники.

Model Sim. Раніше існувала фірма Mentor Graphics випустила симулятор, що використовується досі, якийсь час з ним можна було ознайомитись через altera.com. Оскільки великі фірми до своїх продуктів включають кілька симуляторів.

У фірми Aldec Active-HDL можна завантажити програму та файли, необхідні для моделювання. Для зручності використання та скачування, є можливість вибрати, ПЛМ і ПЛІС, яких саме фірм бажає завантажити користувач. У переліку виробників – Altera, LatticeSemi, Xilinx.

Висновки. Таким чином, можна здійснити вибір програмного забезпечення для моделювання. Щоправда, підтримка апаратного забезпечення буде неповною. Лише моделювання.

Список використаних джерел.

1. https://en.wikipedia.org/wiki/Hardware_description_language
2. ChatGPT Aria for Opera <https://www.opera.com/ua/features/browser-ai>.
3. ModetSim <https://eda.sw.siemens.com/en-US/ic/modelsim/>
4. Aldec Active-HDL https://www.aldec.com/en/products/fpga_simulation/active_hdl_student
5. I. Svyd, O. Vorgul, V. Semenets, O. Zubkov, V. Chumak, N. Boiko. Special Features of the Educational Component “Design of Devices on Microcontrollers and FPGA”. // II International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs (MC&FPGA), Kharkiv, Ukraine, 2020, pp. 55-57. doi: 10.35598/mcfpga.2020.017
6. O. Vorgul, I. Svyd, O. Zubkov, V. Semenets. Teaching microcontrollers and FPGAs in Quarantine from Coronavirus: Challenges and Prospects. // II International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs (MC&FPGA), Kharkiv, Ukraine, 2020, pp. 14-17. doi: 10.35598/mcfpga.2020.005
7. I.V. Svyd, O.V. Litvinenko, O.G. Bilotserkivets. Features of designing digital devices based on Xilinx FPGA in CAD Vivado HLx Design Suite. // Specialized Exhibition "KharkivProm Days. Production and efficiency". Collection of materials of the forum section "Automation, electronics and robotics. Development Strategies and Innovative Technologies". - Kharkiv, KNURE, Exhibition Company ADT, 2019, pp. 43-44.
8. В. Чумак, І. Свид. Створення модуля VHDL-опису при проектуванні цифрових систем на ПЛІС в Xilinx ISE Design Suite. // Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2019). – Дніпро, Дніпровський національний університет імені Олеся Гончара, Кременчук: ПП Щербатих О. В., 2019. – С. 94-95.