

## РОЗРОБКА ПРОГРАМНО-ВИЗНАЧУВАНИХ РАДІОСИСТЕМ НА БАЗІ SoC В СЕРЕДОВИЩІ MATLAB/SIMULINK

Бондаренко К.С., Кайдено М.М., Роскошний Д.В.

Науково-дослідний інститут телекомунікацій

КПІ імені Ігоря Сікорського, Україна, Київ

E-mail: kateolena@gmail.com, kkk610@ukr.net, ddd610@ukr.net

### Development of the Software-Defined Radio based on SoC in the Matlab / Simulink Environment

This article describes the process of creating a program code in the Matlab/Simulink environment and further compilation of created code in specialized environments for development and design of a SoC-based SDR

Розробка програмно-визначуваних радіосистем (SDR - Software-Defined Radio) на базі технологій програмовані системи на кристалі (SoC – FPGA+HPS) це складний та трудомісткий процес, який потребує використання декількох середовищ для прототипування, розробки та тестування таких систем. В роботах [1,2], було описано процес модельно-орієнтованого проектування для створення програмно визначуваних систем з використанням середовища Matlab/Simulink. Слід зазначити, що SDR системи в тому числі і на основі SoC, відносяться до класу вбудованих систем (embedded system) і для їх проектування використовуються методи та засоби, які в загальному вигляді відрізняються від тих, що застосовуються для проектування комп'ютерних систем загального призначення. Дана робота присвячена опису процесу створення програмного коду для SoC з середовища Matlab/Simulink для подальшої компіляції в спеціалізованих середовищах для розробки та проектування. На рис.1 показано структуру процесу створення радіосистеми з використанням модельно-орієнтованого проектування.

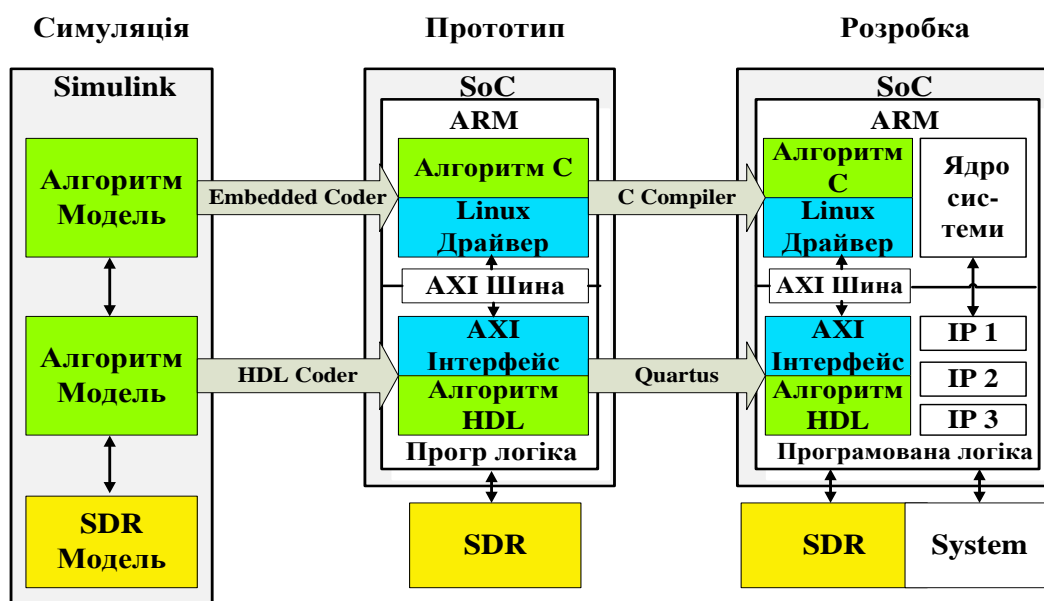


Рис.1. Структура процесу створення програмно-визначуваної радіосистеми з використанням модельно-орієнтованого проектування.

В середовищі Matlab/Simulink для переходу від процесу симуляції до прототипу використовуються як вбудовані засоби, так і засоби сторонніх розробників (third-party software).

Створення прототипу для FPGA полягає в створенні коду на мові verilog/VHDL. Для цього в середовищі Matlab повинні були встановлені наступні компоненти:

- *HDL Coder* для генерації verilog чи VHDL коду з функцій MATLAB, Simulink моделей чи Stateflow charts;
- *Fixed-Point Designer*, який підтримує типи даних і надає інструменти для розробки алгоритмів з фіксованою точкою одинарної точності для оптимізації продуктивності embedded system;
- Пакет підтримки для цільової платформи, наприклад *HDL Coder Support Package for Altera SoC Platform*, *HDL Coder Support Package for Xilinx Zynq Platform*, та ін.;
- Середовище розробки FPGA в залежності від SoC, який буде використовуватись в цільовій платформі: Altera Quartus II, Xilinx ISE, Xilinx Vivado.

Для підключення відповідного середовища в командному вікні Matlab виконується команда `hdlsetuptoolpath('ToolName', 'Назва середовища', 'ToolPath', 'шлях до файлу запуску середовища')`.

Процес створення прототипу здійснюється з використанням *HDL Workflow Advisor* для вибраної підсистеми, графічне зображення якого показано на рис. 2 [3] При встановленні параметрів *HDL Workflow Advisor* для SoC встановлюється режим генерації IP core для вибраної цільової платформи і відповідно SoC-пристрою. Вид графічного вікна *HDL Workflow Advisor* показано на рис.2.

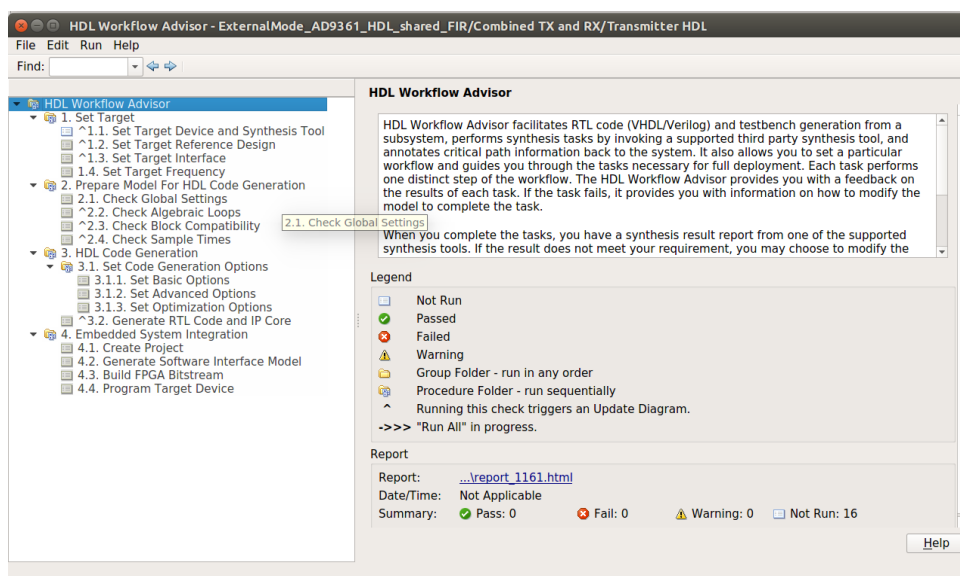


Рис.2 Графічне вікно *HDL Workflow Advisor* для створення коду на мові verilog підсистеми Transmitter HDL

В процесі створення коду у випадку роботи в операційній системі Windows можливе створення моделі інтерфейсу програмного забезпечення з блоками драйверів IP Core для генерації коду C (4.2. Generate Software Interface Model). В середовищі Linux така можливість відсутня в наслідок того, що в

Matlab відсутні пакети підтримка генерації вбудованого коду для ARM-процесорів відповідних цільових SoC (Embedded Coder Support Package for ARM Cortex-A Processors).

Процес створення прототипу закінчується (за необхідності внутрішньсхемної симуляції FIL — FPGA in the Loop) синтезом та генерацією прошивки (bitstream) для вбудованої системи FPGA та її програмуванням через порти JTAG, Ethernet, або записом на носій. Згенерований проект IP Core може оптимізуватись та модернізуватись вже безпосередньо в середовищі розробки FPGA.

Створення прототипу для ARM процесора SoC полягає в створенні коду на мові C/C++. Для цього в середовищі Matlab повинні були встановлені наступні компоненти:

- Embedded Coder для генерації C та C++ коду, який оптимізований для вбудованих систем;
- DSP System Toolbox — пакет цифрової обробки сигналів;
- DSP System Toolbox Support Package for ARM Cortex-A Processors;
- Embedded Coder Support Package for ARM Cortex-A Processors;
- Embedded Coder Support Package for ARM Cortex-M Processors.

Внаслідок того, що в середовищі Linux відсутня можливість використовувати Embedded Coder Support Package for ARM Cortex-A Processors замість нього може бути використаний Embedded Coder Support Package for ARM Cortex-M Processors, хоча цей варіант і не є оптимальним. Іншим, більш оптимальним варіантом є використання програмного пакету DS-5 разом з програмною оболонкою Eclipse. Для роботи з DS-5 необхідна інсталяція до середовища Matlab C/C++ Toolchain for Arm processors [4] Цей інструмент забезпечує підтримку всіх типів ARM процесорів як для середовища Linux, так і для середовища Windows.

Процес створення прототипу здійснюється з використанням Embedded Coder Quick Start. Попередньо до запуску Embedded Coder Quick Start необхідно встановити параметри генерації коду, як показано на рис. 3, вибравши необхідний системний *Target* файл, мову програмування та *Toolchain*, а також інші параметри. Окрім того доцільно запустити Code Generation Advisor, за допомогою якого встановлюються цілі генерації коду та здійснюється перевірка моделі. Embedded Coder Quick Start виконує генерацію коду для моделі, або для підсистеми в межах моделі. Коли модель містить інші підсистеми, які не призначені для генерації коду, вибирається тільки одна підсистему для збірки коду.

В результаті створюється прототип на мові C/C++ включаючи Makefile необхідний для компіляції. Після створення прототипу перенесення на цільову платформу здійснюється шляхом крос-компіляції. Не зважаючи на те, що в середовищі Windows процес створення прототипу для ARM Cortex-A є простішим, більш доцільним є робота в середовищі Linux. Це є особливо важливим при використанні в цільовій платформі Embedded Linux в силу наступних причин:

- при крос-компіляції з Windows до Linux виникають проблеми залежностей та бібліотек, що підключаються динамічно;

- інструменти в каталозі джерел Linux./scripts напряму не сумісні з середовищем Windows і, таким чином, хоча вони можуть компілюватись, вони не працюють належним чином і повинні бути виправлені в ручному режимі;
- Matlab генерує Makefile для моделі Simulink з ім'ям моделі name\_model.mk з врахуванням цільової платформи, але при крос-компіляції цей файл може не враховувати багатьох можливостей і повинен бути виправлений в ручному режимі.

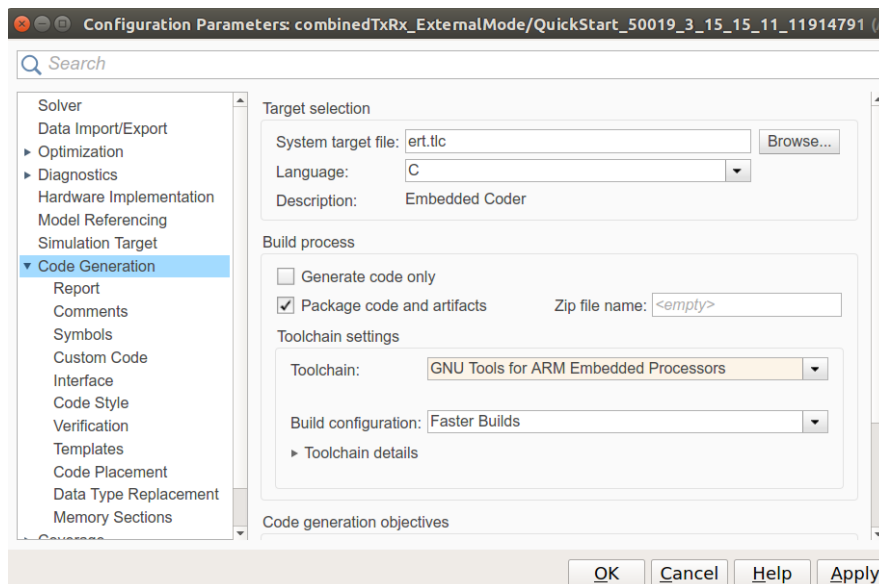


Рис. 3. Графічне вікно *Code Generation Option* для встановлення параметрів генерації коду

Процес крос-компіляції залежить від середовища, в якому вона здійснюється, в роботі [5] представлено процес крос-компіляції, проблеми та способи її вирішення при створенні машинного коду з використанням режиму командної строки. Компіляція здійснювалась в середовищі Linux (Ubuntu 16.04) Використовувався крос-компілятор для bare metall application *gcc-arm-none-eabi*, аналогічно для embedded Linux може бути використаний компілятор *gcc-arm-linux-gnueabi*.

### Література

1. Кайденко Н.Н., Кайденко В.Н. «Программно-аппаратное моделирование телекоммуникационных процессов и систем в MATLAB с использованием SDR и SOC технологий» // X Міжнародна науково-технічна конференція "Проблеми телекомунікацій" ПТ-2016: Збірник матеріалів конференції. К.: НТУУ "КПІ", 2016. - с. 250-253.
2. Кайденко М.М. «Використання модельно-орієнтованого проектування для програмно-визначуваних радіосистем» // XI Міжнародна науково-технічна конференція "Проблеми телекомунікацій" ПТ-2017: Збірник матеріалів конференції. К.: КПІ ім. Ігоря Сікорського, 2017 с. 181-183
3. Getting Started with Hardware-Software Co-Design Workflow for Intel SoC Devices - Режим доступу: <https://www.mathworks.com/help/hdlcoder/examples/getting-started-with-hardware-software-codesign-workflow-for-intel-soc-devices.html>
4. Arm Compiler - Режим доступу: [https://www.mathworks.com/products/connections/product\\_detail/arm-compiler.html](https://www.mathworks.com/products/connections/product_detail/arm-compiler.html)
5. How to: build module MATLAB cross-compilation - Режим доступу: <https://bitbucket.org/mnacakanov/tropo/wiki/How%20to%20build%20module%20MATLAB%20cross-compilation>