

Генератор QR-коду на основі FPGA

Сокол^f Я. В., ORCID [0000-0002-9104-6690](https://orcid.org/0000-0002-9104-6690)

Варфоломеев^s А. Ю., к.т.н., ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Київ, Україна

Анотація—Розроблено апаратне IP-ядро для генерування QR-кодів на ПЛІС, що підтримує першу версію кодів з корекцією помилок М типу. На основі розробленого IP-ядра, на платі налагодження DE1-SoC створено макетний зразок пристрою для генерування QR-кодів. Зазначений пристрій дозволяє за допомогою клавіатури з інтерфейсом PS/2 вводити текстові дані, які мають бути зашифровані у QR-коді та виводити результат (згенерований код) на VGA монітор у вигляді зображення. Для створення такого пристрою використовується модуль керування, IP-блок для генерування QR-кодів та модулі введення-виведення.

Ключові слова — QR-код; FPGA; VGA; клавіатура.

I. ВСТУП

QR-код (Quick Response) є двомірним матричним штрих-кодом, який призначений для зберігання невеликих обсягів інформації, головним чином на друкованих носіях, яку можна швидко та зручно зчитати і розпізнати як за допомогою спеціальних сканерів, так і камерою мобільного телефону. В загальному випадку у QR-коді можна закодувати будь-яку інформацію, проте найбільш часто у ньому зберігають текстову інформацію, таку як номери телефонів, посилання на сайти, геометки, номери криптовалютних гаманців тощо [1, 2].

Не дивлячись на існування альтернативних версій двомірних штрих-кодів, таких як Aztec code, Data Matrix, саме QR-коди наразі набули найбільшого розповсюдження, тому в цій роботі мова йдеться саме про них.

Створення QR-коду може здійснюватись програмно — із використанням мікропроцесора чи мікроконтролера або апаратно — за допомогою спеціалізованого пристрою. Кожен з цих підходів має свої переваги та недоліки. Так, програмні реалізації є більш гнучкими і простими, водночас, будучи розгорнутими на неспецифічних контролерах чи процесорах, вони потребують більшого енергоспоживання [3]. Апаратні реалізації навпаки є більш складними, не володіють значною гнучкістю, проте можуть бути більш швидкодіючими та споживати значно менше енергії, що робить їх використання більш доцільним у різноманітних портативних пристроях з батарейним живленням. При цьому, щоб спростити створення апаратного модуля для генерування QR-коду доцільно скористатися технологією ПЛІС.

Таким чином, мета даної роботи полягає в розробці вбудованого рішення для генерування QR-кодів

на ПЛІС. Для цього було розроблено спеціальний IP-блок на базі плати налагодження DE1-SoC.

II. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ

IP-блок для генерування QR-кодів пропонується реалізувати на доступній та поширеній платі налагодження DE1-SoC, що містить FPGA мікросхему Cyclone V. Дана плата має інтерфейси для підключення клавіатури та VGA — монітору, тому введення та виведення інформації про QR-код доцільно реалізувати через зазначені периферійні пристрої. Пропонується наступний принцип роботи розроблюваного пристрою:

- користувач вводить необхідний текст з клавіатури; даний текст паралельно відображається на моніторі;
- при натисканні на кнопку “Enter” виконується генерування QR-коду та його виведення у вигляді зображення на монітор;
- за необхідності повторне введення коду здійснюється при натисканні на клавішу “Esc”.

Загальна структурна схема пристрою та структурна схема IP-блоку генерування QR-кодів представлені на Рис. 1 а) та б) відповідно.

IP-блок для генерування QR-кодів реалізовано на мові Verilog [4] і складається з наступних модулів: `control`, `keyboard`, `mem_gen_code`, `mem_main_code`, `mem_mat_code`, `VGA`. Для спрощення алгоритму використовується обмеження, які полягають в наступному:

- Генерування здійснюється лише для першої версії коду, яка має розмір матриці 21x21 піксель.



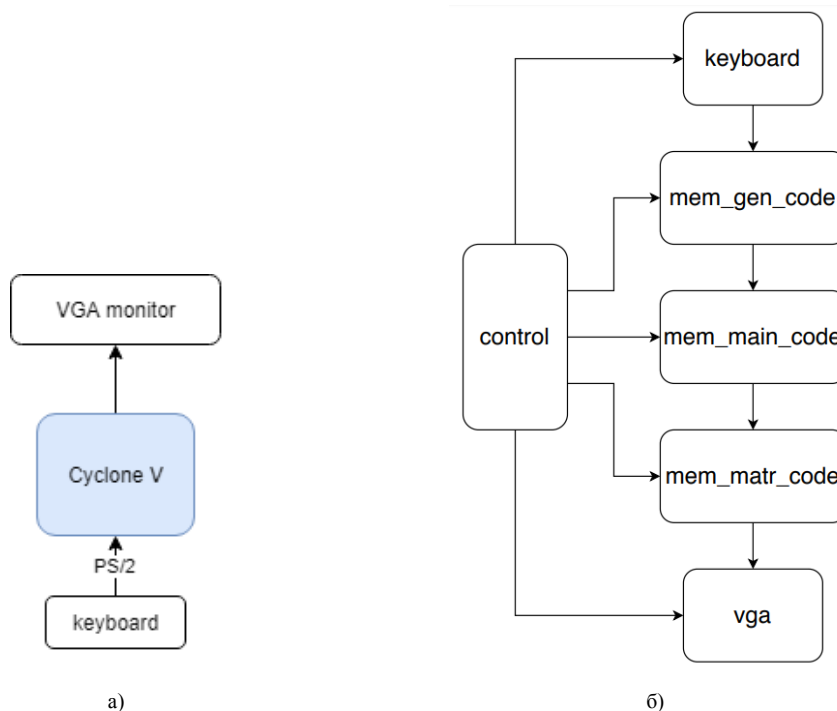


Рис. 1 Структурна схема пристрою та IP-блоку генерування QR-кодів: а) загальна структурна схема пристрою; б) Структурна схема IP-блоку генерування QR-кодів

- Застосовується рівень корекції помилок М типу. Даний тип корекції передбачає використання у кодї надлишкової інформації, що дозволяє відновити вміст закодованих даних у випадку пошкодження до 15% зображення коду (такий тип є найбільш розповсюдженим).

Виходячи з даних обмежень, код може шифрувати не більше ніж 14 символів.

III. ОПИС РЕАЛІЗАЦІЇ АПАРАТНИХ МОДУЛІВ ПРИСТРОЮ ГЕНЕРУВАННЯ QR-КОДІВ

Апаратна реалізація пристрою для генерування QR-код на платі налагодження DE1-SoC включає модуль керування, IP-блок для генерування коду та модулі введення-виведення. Нижче наведемо їх опис на високому рівні.

A. Модуль керування

Керування пристроєм здійснюється за допомогою модуля **control**, який є скінченним автоматом зі станами, які відповідають наступним операціям:

- скидання, очищення пам'яті;
 - запис коду з клавіатури;
 - створення основної частини QR-коду;
 - створення байтів корекції використовуючи на основі даних основної частину QR-коду;
 - об'єднання байтів корекції з основною частиною QR-коду та заповнення їх у матрицю 21x21 піксель;
- виведення коду на екран;
 - очікування кнопки Esc

B. IP-блок для генерування QR-кодів

Блок складається з трьох модулів **mem_main_code**, **mem_gen_code** та **mem_matr_code**. Наведемо більш детальний їх опис.

Модуль **mem_main_code** приймає на вхід байти тексту у форматі ASCII [5] і генерує первинну послідовність даних, що мають бути зашифровані у кодї. Ця послідовність містить наступні поля:

- 1) Перші 4 біти зберігають тип кодування. В даній роботі використовується лише побайтове кодування, тому дане поле містить послідовність 0100. Загалом існують ще цифрове та літерно-числове кодування, для яких перше поле має заповнюватись бітами 0001 та 0010 відповідно.
- 2) Наступні 2 байти зберігають обсяг інформації, що зашифрована в кодї — це звичайне двійкове число.
- 3) Наступні N байт (кількість залежить від типу корекції та номеру версії) зберігають корисну інформацію (власне дані). В даній роботі використовується до 14 байтів даних. Невикористані байти заповнюються наступними константами по черзі: 11101100 та 00010001.
- 4) Останні 4 біти заповнюються нулями для того, щоб послідовність містила цілу кількість байт.

Модуль `mem_gen_code` виконується для генерування байтів корекції. Для цього виконується алгоритм Ріда-Соломона [6].

Для застосування алгоритму Ріда-Соломона необхідно для початку підготувати масив байтів корекції. Даний масив обирається за рівнем корекції та номером версії коду. Оскільки використовується рівень корекції M та перший номер версії, то масив байтів корекції містить наступні 10 елементів: {251, 67, 46, 61, 118, 70, 64, 94, 32, 45}. Для даного алгоритму також необхідні таблиці прямого та зворотного полів Галуа, які зберігають значення від 0 до 255 [7]. Сам алгоритм передбачає циклічне виконання наступні дії (кількість ітерацій циклу визначається кількістю байт даних, що мають бути закодовані) [6]:

- 5) перший елемент масиву даних зберігається у змінній a і видаляється з масиву (всі наступні значення зсуваються на одну клітинку вліво, останній елемент заповнюється нулем);
- 6) якщо a дорівнює нулю, то пропустити наступні дії і перейти до наступної ітерації циклу;
- 7) у таблиці зворотного поля Галуа визначається значення комірки, що відповідає числу a і її значення заноситься у змінну b : $b = GF^{-1}[a]$;
- 8) далі для N перших елементів, де N — кількість байтів корекції, i – лічильник циклу;
- 9) до i -го значення генеруючого многочлена додається значення b і зберігається у змінній c ;
- 10) якщо b більше 254, використовується залишок від ділення b на 255;
- 11) знайти відповідне c значення в таблиці поля Галуа і виконати побітову операцію сума по модулю 2 (XOR) з i -м значенням підготованого масиву і записати отримане значення в i -у комірку підготовленого масиву.

Модуль `mem_matr_code` виконує генерування самої матриці, розміщуючи узори для вирівнювання, код маски та рівень корекції та смуги синхронізації відповідно з Рис. 2 [8].

Дана інформація містить наступне [6]:

- Узори для вирівнювання завжди розташовуються в 3 кутах (верхній лівий кут, верхній правий кут та нижній лівий кут). Матриця коду також може містити додаткові узори для вирівнювання, що розміщуються через кожну 21 комірку матриці по висоті та ширині.

Додаткові узори наявні у версіях QR-кодів вище 2-ї, тому в цій роботі не застосовуються.

- Тип маски та рівень корекції задаються 15-бітним числом. Оскільки в даній роботі використовується рівень корекції типу M та тип маски під номером 1, то код типу маски та рівня корекції містить бітову послідовність наступного вигляду 101000100100101.
- Полоси синхронізації розміщуються між узорами для вирівнювання і почергово заповнюються нулями та одиницями.
- Номер версії коду складається з 15 бітів, обирається згідно номеру версії QR-коду і використовується в 7 версії та більше.

Далі після цього виконується заповнення байтів з певною послідовністю, які були згенеровані за допомогою блоку `mem_main_code`, відповідно до Рис. 3 [6]. Після такого розташування для цих даних накладається маска, відповідно до Рис. 4 [9] (використовується маска 1-го типу, її код 001).

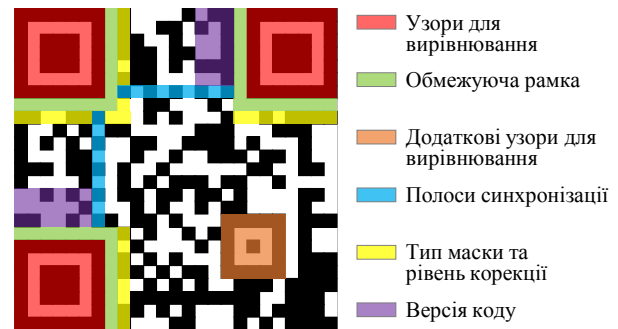


Рис. 2 Розміщення службової інформації на QR-коді

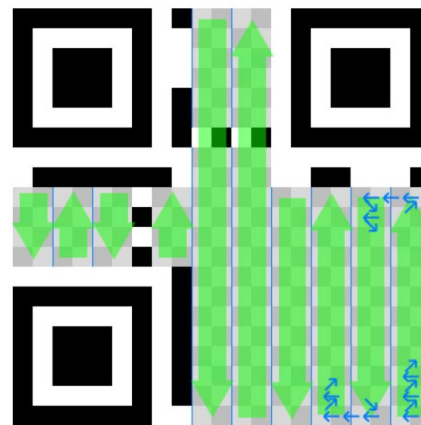


Рис. 3 Розміщення даних на QR-коді

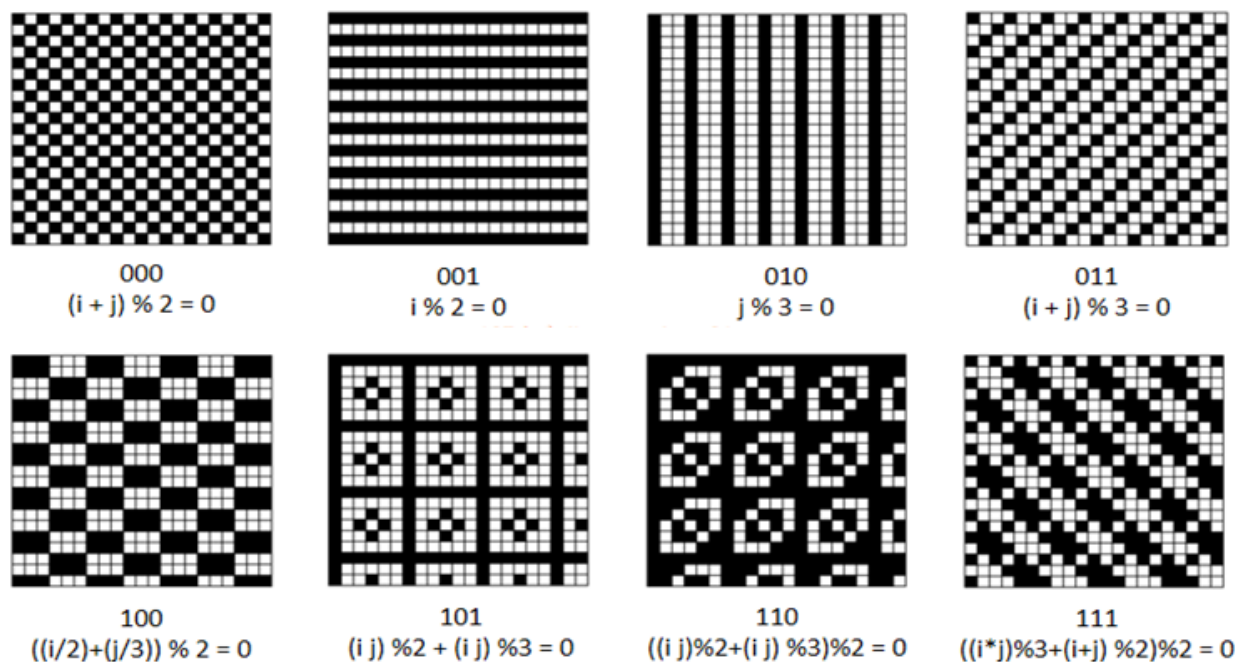


Рис. 4 Типи масок: операція «%» позначає остачу від ділення

С. Модулі введення-виведення

Модуль keyboard виконує зчитування сигналів клавіатури (сигналу синхронізації та сигналу даних) і перетворення сигналу даних на ASCII-коди. Для цього по кожному задньому фронту сигналу синхронізації виконується реєстрація значень сигналу даних, які далі записуються у 8-бітну шину. Оскільки клавіатура може мати «брязкіт» контактів на клавішах, в даному модулі додатково виконується фільтрація прийнятих даних. Фільтрація «брязкоти» контактів виконується за допомогою затримки: як тільки отримується перше прийняття даних з клавіатури створюється затримка, під час якої ігнорується повторне прийняття одних і тих же даних.

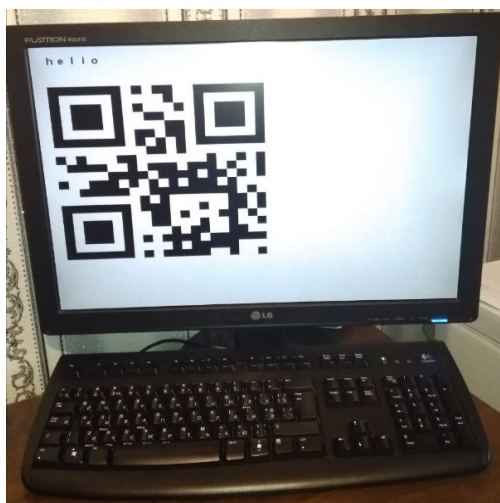


Рис. 5 Результат генерування QR-коду

Модуль VGA формує необхідні синхросигнали для правильної роботи VGA монітору [10]. Після генерації QR-коду виконується його збільшення масштабу (дана збільшена матриця передається на один з вихідних сигналів). Також, паралельно з друком тексту виконується створення матриці пікселів, які відображають текст і передається на вихідний сигнал.

Результат роботи створеного генератора QR-коду показано нижче на Рис. 5.

ВИСНОВКИ

Розроблено апаратний IP-блок для генерування QR-кодів на ПЛІС. На основі даного блоку на платі налагоджування DE1-SoC реалізовано апаратний пристрій, що дозволяє вводити текстову інформацію з клавіатури, генерувати з неї QR-код та виводити його на VGA монітор. Створений пристрій підтримує лише QR-коди першої версії та наразі може кодувати до 14 символів даних, застосовуючи побайтове кодування та рівень корекції типу M (дозволяє відновлювати інформацію із зображення, яке пошкоджено на 15%).

Враховуючи наявність обмежень маємо новий пристрій. За рахунок даних обмежень пристрій хоч може зберігати меншу кількість даних, але має більшу швидкість генерування QR-коду.

Враховуючи наявні обмеження у поточній реалізації пристрою, основою для подальшого його покращення можуть стати:

- збільшення довжини кодованих даних;
- підтримка інших методів кодування (цифрового та побайтового);
- додати наступні рівні корекції: L для 7%, Q для 15% та H для 30% пошкоджень;



- можливість використання інших масок.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] "Generator QR codov [Генератор QR кодів]," [Online]. Available: <http://qrcoder.ru/>. [Accessed 15 03 2020].
- [2] "Pochemu nam stoit prismetret'sya k QR-kodam v 2019 godu [Почему нам стоит присмотреться к QR-кодам в 2019 году]," [Online]. Available: <https://lifehacker.ru/qr-kody-v-2019-godu/>. [Accessed 15 03 2020].
- [3] "PLIS (FPGA) i mikrokontroller. V chem raznitsa? [ПЛИС (FPGA) и микроконтроллер. В чем разница?]," [Online]. Available: <http://micro-proger.ru/2016/03/17/plis-fpga-i-mikrokontroller-v-chem-raznica/>. [Accessed 15 03 2020].
- [4] "Verilog Language ," [Online]. Available: <https://en.wikipedia.org/wiki/Verilog>. [Accessed 15 03 2020].
- [5] "ASCII," [Online]. Available: <https://en.wikipedia.org/wiki/ASCII>. [Accessed 15 03 2020].
- [6] "Algoritm generatsii QR-koda [Алгоритм генерации QR-кода]," [Online]. Available: <https://habr.com/ru/post/172525/>. [Accessed 15 03 2020].
- [7] K. Kasperski, "Polynomial'naya arifmetika i polya Galua, ili Informatsiya, voskresshaya iz pepela [Полиномиальная арифметика и поля Галуа, или информация, воскрешая из пепла]," *Sistemnyy administrator*, p. 84–90, 2003.
- [8] "Wounded QR codes," [Online]. Available: <http://datagenetics.com/blog/november12013/index.html>. [Accessed 15 03 2020].
- [9] "Algoritmy raspoznavaniya graficheskogo markera [Алгоритмы распознавания графического маркера]," [Online]. Available: https://studbooks.net/2238811/informatika/algoritmy_raspoznavaniya_graficheskogo_markera. [Accessed 15 03 2020].
- [10] *Terasic DE1-SoC User Manual*, 2014, p. pp. 30–33.

UDC 621.389

QR-Code Generator Based on FPGA

Ya. V. Sokol^f, ORCID [0000-0002-9104-6690](https://orcid.org/0000-0002-9104-6690)A. Yu. Varfolomeiev^s, PhD, ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Department of Design of Electronic Computing Equipment, Faculty of Electronics
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
Kyiv, Ukraine

Abstract—The IP-core for QR-code generation, which supports codes of the first version and the M-type level of correction (allows to recover information from damaged up to 15% code image) is developed. Based on this IP-core and the DE1-SoC development board, containing the Cyclone V FPGA chip, the real QR-code generation device was implemented. The device uses the PS/2 interface to enter text data that should be encoded in the QR-code and output the result (generated code) on a VGA monitor as an image. The suggested device has the following structure: the control module, the IP-core for QR-code generation and I/O modules. The control module performs the general device management and implemented as the state machine, which has the following states: reset and memory cleaning, read the text that should be encoded from the keyboard, create the main part of the QR-code, create correction data, combine the correction data with the main part of the QR-code, transform the result into a 21x21 pixel matrix and displaying the code on the screen. The input module reads the keyboard signals (synchronization and data signals) and converts them into the ASCII codes. To do this, on each falling edge of the synchronization signal the values of the data signal are registered and propagated to the 8-bit bus. Since the keyboard may have a contact bounce, this module additionally filters the received data using the delay filtering. The output module is the VGA controller that simply generates the necessary clock signals for the proper VGA monitor functioning. After the QR-code is generated, its matrix is enlarged and passed to the VGA controller. Also, in parallel with that, the text, which was encoded in the code, is printed on the display. The IP block for QR-code generation consists of three submodules. The first one receives a text in ASCII format (now only 14 characters are supported) and generates the primary sequence of data to be encrypted in the code. The second submodule calculates the correction data using the Reed-Solomon algorithm. To use this algorithm, the additional array is reserved to store the correction data. This array is filled with special information, depending on the level of correction and the code version number. The algorithm also requires tables of the forward and reverse Galois fields, which are precalculated and stored in the submodule's ROM. The third module generates the matrix itself: it places the alignment patterns, the information of the used type of mask, level of correction, and adds timing patterns. After that, the mask superimposed on the data generated by the previous module, and the result is copied to the matrix code in the required order.

Keywords — QR-code; FPGA; VGA; keyboard.

