**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

-----🕮-----

# THIẾT KẾ MẠCH ĐIỆN VÀ ĐÁNH GIÁ MỨC ĐỘ BẢO MẬT CỦA BỘ SINH SỐ NGẪU NHIÊN TRÊN CÁC LINH KIỆN RỜI RẠC VÀ TÍCH HỢP VÀO HỆ THỐNG VI XỬ LÝ MICROBLAZE


| | | |
|---|---|---|
| Giảng viên: | GS.TS Trần Xuân Tú | |
| | TS. Bùi Duy Hiếu | |
| Nhóm sinh viên: | Phạm Thành Nam | 20021251 |
| | Phạm Thị Huyền Trang | 20021271 |
| | Võ Tá Phong | 20021258 |
| | Bùi Thị Quỳnh Nga | 20021252 |


**Hà Nội - 2023**

## TÓM TẮT

**Tóm tắt:** Bộ sinh số ngẫu nhiên thực - True Random Number Generator (TRNG) là một trong những giải pháp quan trọng trong bảo mật. Độ phổ biến của TRNG đã tăng lên do nhu cầu ngày càng cao trong các ứng dụng an ninh mạng, mã hóa dữ liệu và các lĩnh vực khác yêu cầu số ngẫu nhiên chất lượng cao. Do đó, việc nghiên cứu về TRNG rất cần được quan tâm và xem xét. Nhận thấy tính cấp bách của đề tài nên nhóm sinh viên chúng em đã tiến hành nghiên cứu và thử nghiệm TRNG. Nội dung của bản báo cáo tập trung trình bày việc thiết kế mạch tạo số ngẫu nhiên dựa trên hiện tượng nhiễu avalanche và kết nối với mạch FPGA. Kiểm thử khả năng tạo số ngẫu nhiên của mạch bằng bài NIST test. Tìm hiểu về các cách tấn công mạch tạo số ngẫu nhiên

***Từ khóa:*** Nhiễu avalanche, FPGA, NIST, TRNG.

## MỤC LỤC

# DANH SÁCH HÌNH ẢNH

# DANH SÁCH CÁC BẢNG

# DANH MỤC TỪ VIẾT TẮT

| Ký hiệu | Tên tiếng anh | Tên tiếng việt |
|---|---|---|
| **DC** | Direct Current | Dòng điện một chiều |
| **FPGA** | Field – Programmable Gate Array | Thiết bị lập trình trường đa cổng |
| **GPIO** | General – Purpose Input/Output | Cổng vào/ra mục đích chung |
| **I/O** | Input/Output | Vào/ra |
| **I2C** | Inter – Integrated Circuit | Mạch tích hợp |
| **IoT** | Internet of Things | Internet vạn vật |
| **IP** | Interllectual Property | Sở hữu trí tuệ |
| **NIST** | National Institute of Standards and Technology | Viện nghiên cứu quốc gia về tiêu chuẩn và công nghệ |
| **RAM** | Random Access Memory | Bộ nhớ truy cập ngẫu nhiên |
| **ROM** | Read – Only Memory | Bộ nhớ chỉ đọc |
| **SDK** | Software Development Kit | Bộ phần mềm phát triển |
| **SPI** | Serial Peripheral Interface | Giao tiếp ngoại vi nối tiếp |
| **TRNG** | True Random Number Generator | Bộ tạo số ngẫu nhiên thực |
| **UART** | Universal Asynchronous Receiver – Transmitter | Bộ truyền nhận nối tiếp không đồng bộ |
| **VHDL** | VHSIC Hardware Description Language | Ngôn ngữ mô tả phần cứng |
| **XDC** | Xilinx Design Constraints | Ràng buộc thiết kế Xilinx |

# CHƯƠNG 1. THIẾT KẾ BỘ TẠO SỐ NGẪU NHIÊN THỰC

## 1.1. Giới thiệu bộ tạo số ngẫu nhiên thực

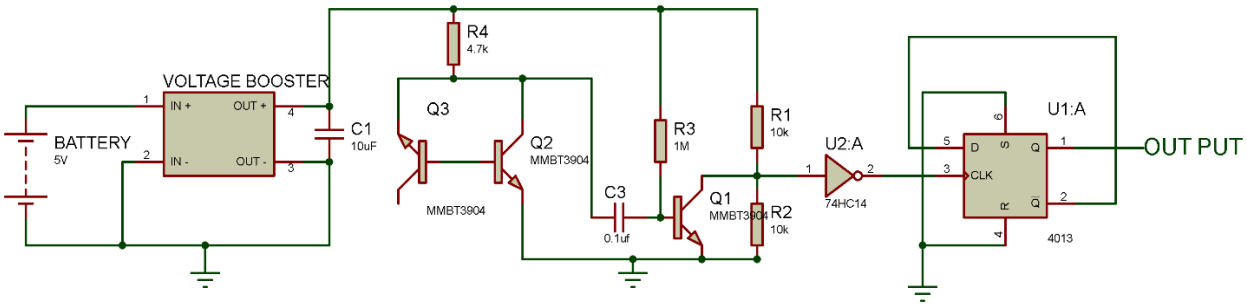
Sự phổ biến của các thiết bị được kết nối và tính chất ngày càng gia tăng của các cuộc tấn công, vi phạm và phần mềm độc hại khiến nhu cầu bảo mật trong các sản phẩm trở nên quan trọng hơn bao giờ hết. Các số ngẫu nhiên thực sự - True random number là trọng tâm của bất kỳ hệ thống bảo mật nào và chất lượng của chúng góp phần nâng cao sức mạnh bảo mật của thiết kế. Ngày nay, số ngẫu nhiên thực sự được yêu cầu quan trọng nhất trong mật mã và vô số ứng dụng của nó vào cuộc sống hàng ngày của chúng ta: thông tin di động, truy cập e-mail, thanh toán trực tuyến, thanh toán không dùng tiền mặt, ngân hàng điện tử, giao dịch qua Internet, điểm bán hàng, thẻ trả trước, khóa không dây, mô phỏng số, nghiên cứu thống kê, thuật toán ngẫu nhiên, xổ số,v.v [1]. Các số ngẫu nhiên yếu hoặc có thể dự đoán được sẽ mở ra cơ hội cho các cuộc tấn công có thể xâm phạm khóa, chặn dữ liệu và cuối cùng là hack các thiết bị cũng như hoạt động liên lạc của chúng.

Bộ tạo số ngẫu nhiên thực - True Random Number Generator (TRNG) là một hàm hay thiết bị dựa trên những hiện tượng vật lý không thể dự đoán được [2], gọi là nguồn entropy, được thiết kế để tạo ra dữ liệu không xác định (ví dụ: chuỗi số) cho các thuật toán bảo mật gốc. Để có hiệu quả, các số ngẫu nhiên phải không thể đoán trước, độc lập về mặt thống kê (không liên quan đến bất kỳ số ngẫu nhiên nào được tạo trước đó) và phân bố đồng đều (xác suất bằng nhau đối với bất kỳ số nào được tạo) [3].

## 1.2. Thiết kế mạch tạo số ngẫu nhiên dựa trên nhiễu avalanche

### 1.2.1. Sơ đồ mạch và nguyên lý hoạt động

Sơ đồ mạch tạo số ngẫu nhiên được tạo từ nguyên mẫu thiết kế có ở hình 1.1 sau đây và được thiết kế và chỉnh sửa lại thông qua phân mềm proteus.

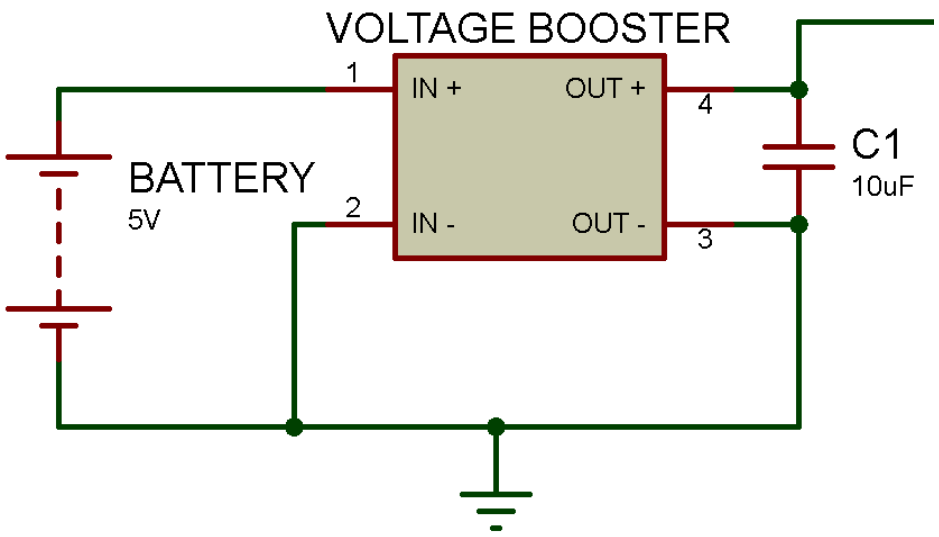



Hình 1.1. Mạch sau khi thiết kế lại bằng phần mềm proteus nguồn [4]

Nguyên lý: Điện áp 5V được cấp vào bộ chuyển đổi tăng áp (Voltage booster), điện áp đầu ra sẽ tăng lên 12V. Dòng điện 12V đi tới transistor Q3 tạo nhiễu và được khuếch đại bởi transistor Q1 và Q2. Bộ chuyển đổi tín hiệu (Hex Schmitt-Trigger inverter) sẽ chuyển đổi các tín hiệu được khuếch thành các tín hiệu xung clock. Khi nhận được xung clock thì D-flip-flop sẽ lưu trữ giá trị trên lối vào D và đưa ra giá trị tương ứng bit 0 hoặc 1 trên lối ra Q. Dữ liệu đầu ra có định dạng là các bit 0 hoặc 1. Đầu ra của mạch sẽ được kết nối với một thiết bị khác để kết nối, hiển thị và ghi lại trên máy tính.

### 1.2.2. Giới thiệu về chức năng của từng khối trong mạch

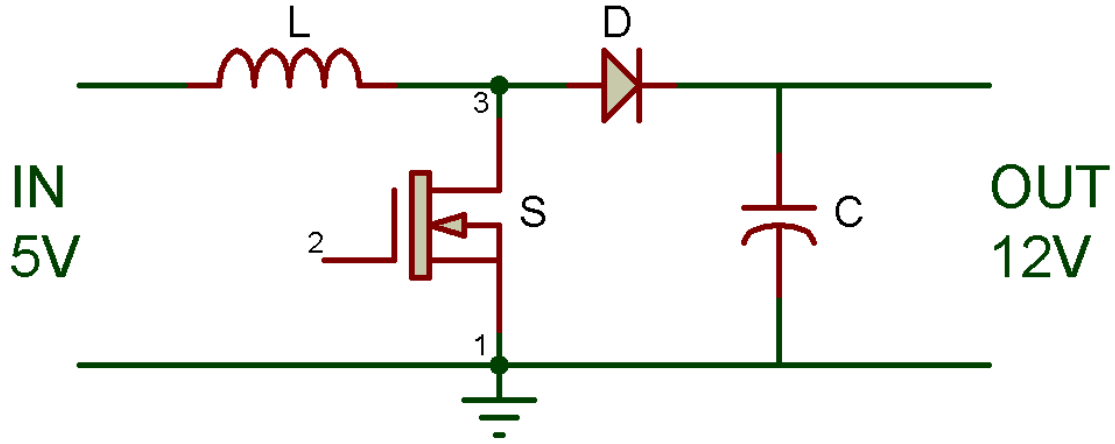
a. Khối tăng áp




Hình 1.2. Khối tăng điện áp

Module tăng áp-boost converter có chức năng tăng điện áp một chiều ở đầu vào thành điện áp một chiều có giá trị cao hơn ở đầu ra. Tụ điện được đặt ở đầu ra của module có chức năng ổn định điện áp đầu ra ở mức 12V.

Cách hoạt động của module tăng áp dựa trên nguyên lý lưu trữ năng lượng trên trong cuộn cảm. Lượng điện áp giảm xuyên suốt của cuộn cảm tỷ lệ thuận với sự thay đổi về dòng điện chạy trong mạch.




Hình 1.3. Sơ đồ mạch của một module tăng áp [5]

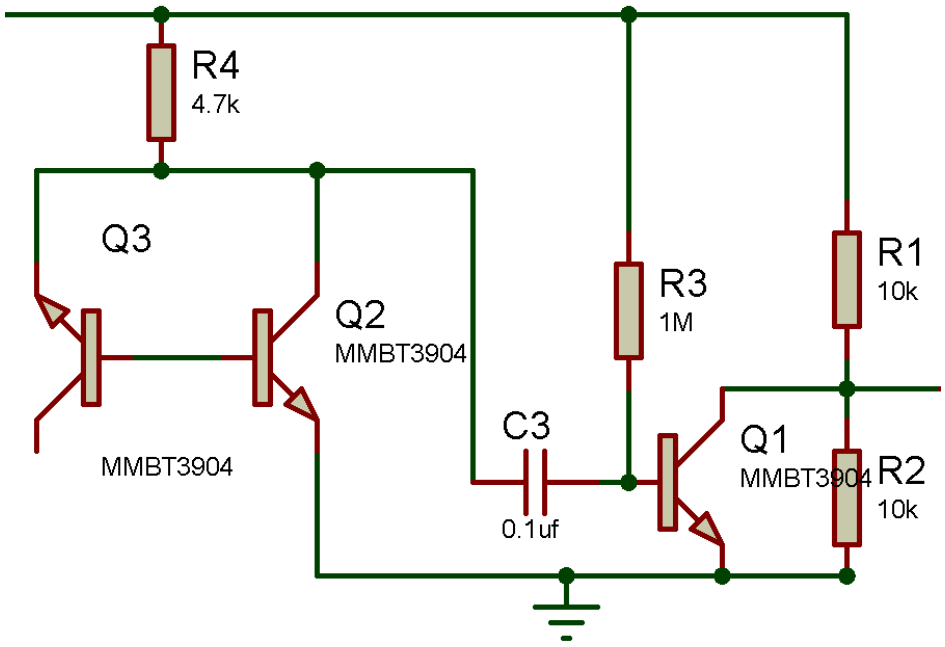
Trong sơ đồ mạch này, transistor hiệu ứng trường - Metal Oxide Semiconductor Field Effect Transistor (MOSFET) S được đặt vào với chức năng như một công tắc đóng và ngắt mạch, diode D được sử dụng như một công tắc thứ 2 ngăn điện áp ngược. Module sẽ hoạt động ở hai chế độ. Ở chế độ 1, công tắc S bật và D tắt, dòng điện sẽ đi vào cuộn cảm và cuộn cảm sẽ lưu trữ năng lượng do ảnh hưởng ở trường điện từ. Ở chế độ 2, công tắc S tắt và D bật, trong chế độ này, cuộn cảm sẽ giải phóng năng lượng

vào tụ điện điện trở của mạch từ đó tăng điện áp đầu ra. Quá trình bật tắt diễn ra liên tục và nhanh không thể quan sát được bằng mắt thường. [5]

## b. Mạch tạo nhiễu

- Giới thiệu về cách hoạt động của mạch

Mạch hoạt động dựa trên nguyên lý nhiễu avalanche tạo bởi transistor Q3. Tín hiệu nhiễu sau khi được tạo ra sẽ được khuếch đại bởi transistor Q1 và Q2.



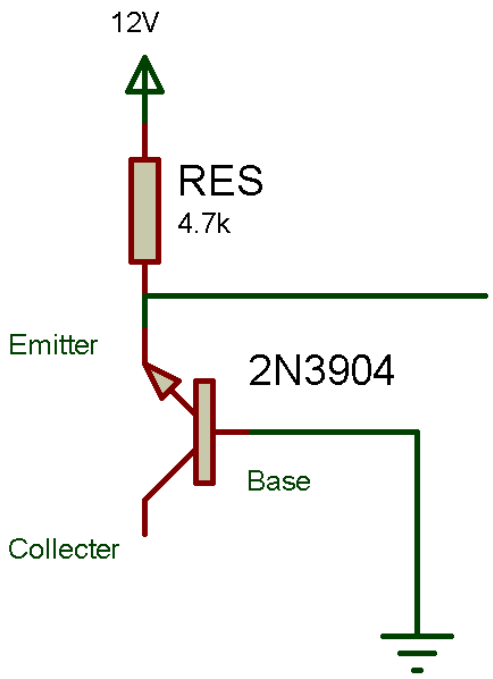

Hình 1.4. Mạch tạo và khuếch đại nhiễu [6]

- Nguyên lý nhiễu avalanche

Khi một điện áp ngược được đặt trên diode vẫn sẽ có một lượng nhỏ dòng điện được đi qua. Khi điện áp ngược tăng lên, nó sẽ chạm tối một mốc khi mà dòng điện tăng đột biến. Sự tăng đột biến của dòng điện dưới tác động của điện áp ngược đó là đặc trưng của sự đánh thủng và giá trị điện áp ngược gây hiện tượng trên được gọi là điện áp đánh thủng.

Nhiễu avalanche là nhiễu được tạo ra khi ta đặt điện áp ngược lên một phân lớp p-n vượt quá điện áp đánh thủng. Nó xảy ra khi bề mặt phân lớp thu được đủ thế năng dưới sự tác động của trường điện tích mạnh tạo ra thêm các cặp điện tích-lỗ trống do sự va chạm của các nguyên tử trong cấu trúc tinh thể. [7]

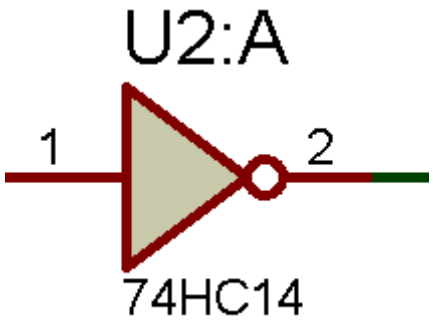
Để có thể tạo ra nhiễu avalanche ta sử dụng phân lớp base-emitter của một NPN transistor bởi vì phân lớp này có điện áp đánh thủng thấp. Lượng nhiễu được tạo ra sẽ phụ thuộc vào tính chất vật lý của phân lớp như vật liệu và mức độ pha tạp. Tại đây, nhiễu avalanche sẽ được ứng dụng làm 1 nguồn entropy. [8]



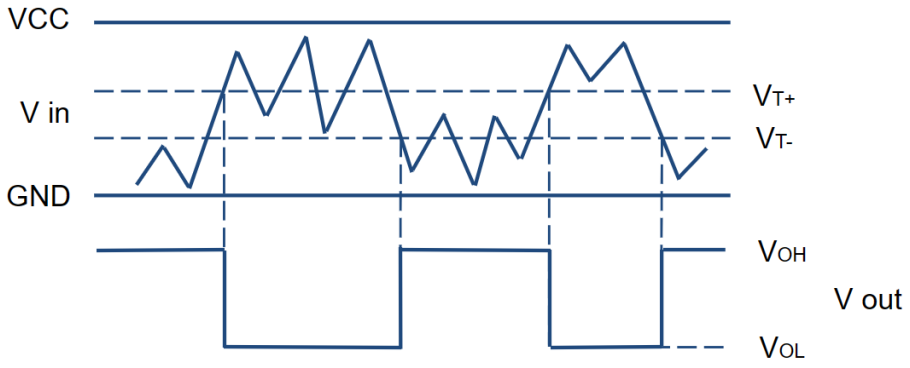

Hình 1.5. Ảnh minh họa của một nhiễu avalanche [9]

c. Khối tạo xung clock

Khối tạo xung clock sử dụng IC 74HC14 nhận điện áp vào là nhiễu sau khi đã khuếch đại. Nếu điện áp đầu vào của khối lớn hơn giá trị VT+ thì điện áp đầu ra sẽ được đặt ở mức thấp VOL. ngược lại, nếu điện áp đầu vào của khối nhỏ hơn giá trị VT- thì điện áp ở đầu ra sẽ được đặt ở mức cao VOH. [10]

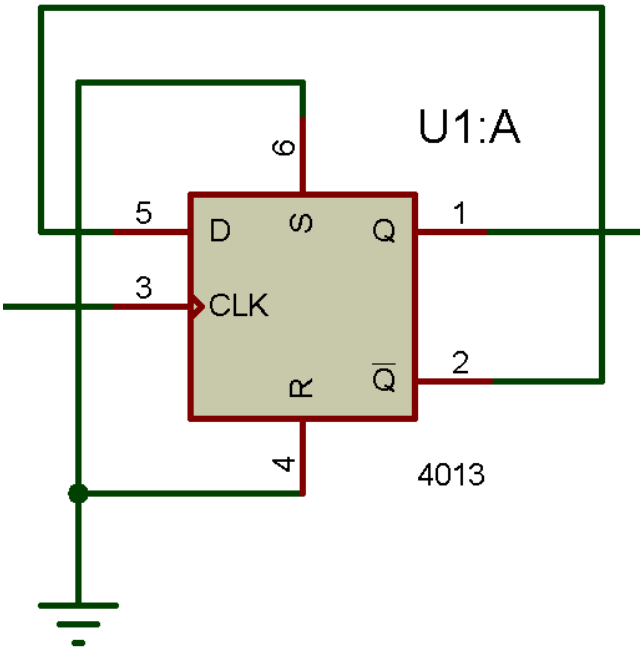



Hình 1.6. IC 74HC14




Hình 1.7. Nguyên lý hoạt động của khối [10]

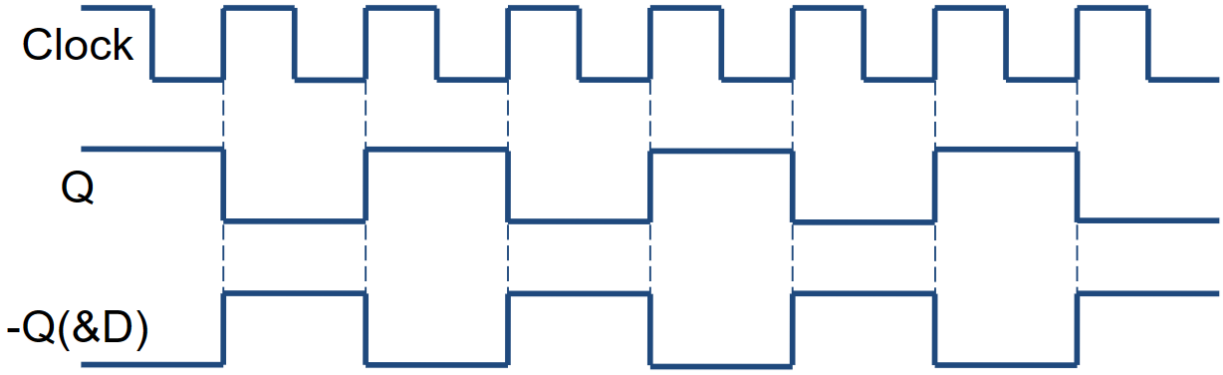
d. Khối D flip-flop




Hình 1.8. Khối D flip-flop

Trong khối flip-flop thì Q là chân dữ liệu ra, $\overline{Q}$ là chân đảo của Q và được kết nối với chân dữ liệu vào D, xung clock sẽ được đưa vào chân CLK của khối.

Khi Q được đặt ở 1 thì $\overline{Q}$ sẽ được đặt ở 0, khi xung clock đầu vào bắt đầu sườn lên thì điện áp Q sẽ được đặt 0 và $\overline{Q}$ được đặt thành 1. Khi xung clock bắt đầu sườn lên tiếp theo lên thì điện áp của hai chân Q và $\overline{Q}$ đảo ngược lại thành 1 và 0. Quá trình diễn ra liên tục và arduino sẽ đọc giá trị đầu ra của chân Q. [11]




Hình 1.9. Nguyên lý hoạt động của khối [11]

## 1.3. Linh kiện thiết kế mạch

Bảng 1.1 dưới đây là danh sách số lượng và thông số của những liên kiện phục vụ cho quá trình thiết kế mạch.

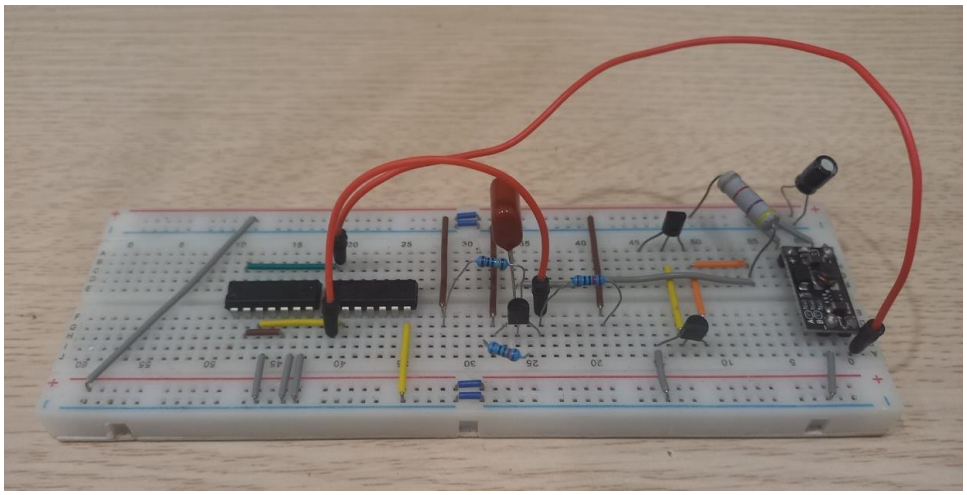
Bảng 1.1. Linh kiện thiết kế mạch

| Tên linh kiện | Mã linh kiện | Số lượng |
|---|---|---|
| Module tăng áp 5V DC lên 12V DC | | 1 |
| Bộ chuyển đổi Hex-Schmitt Trigger | SN74LS14N | 1 |
| D-flop | CD4013BE | 1 |
| Tụ 0.1 microFarad | 0.1uF Cap | 1 |
| Tụ 10 microFarad | 10uF CP | 1 |
| Transistor loại NPN | 2N3904 | 3 |
| Điện trở 4.7K Ohm 1/4W 1% | 4.7k resistors | 1 |
| Điện trở 1M Ohm | 1M resistors | 1 |
| Điện trở 10k Ohm | 10k resistors | 2 |

| | | |
|---|---|---|
| Bảng mạch | Breadboard | 1 |
| Dây nối | Jumpwire | |

## 1.4. Lắp ráp mạch

Sau khi chuẩn bị và kết nối các linh kiện lại với nhau đúng theo như mô phỏng bên trên cho ra kết quả là mạch vật lý như hình 1.10.



Hình 1.10. Mạch vật lý sau khi kết nối

## 1.5. Kết luận

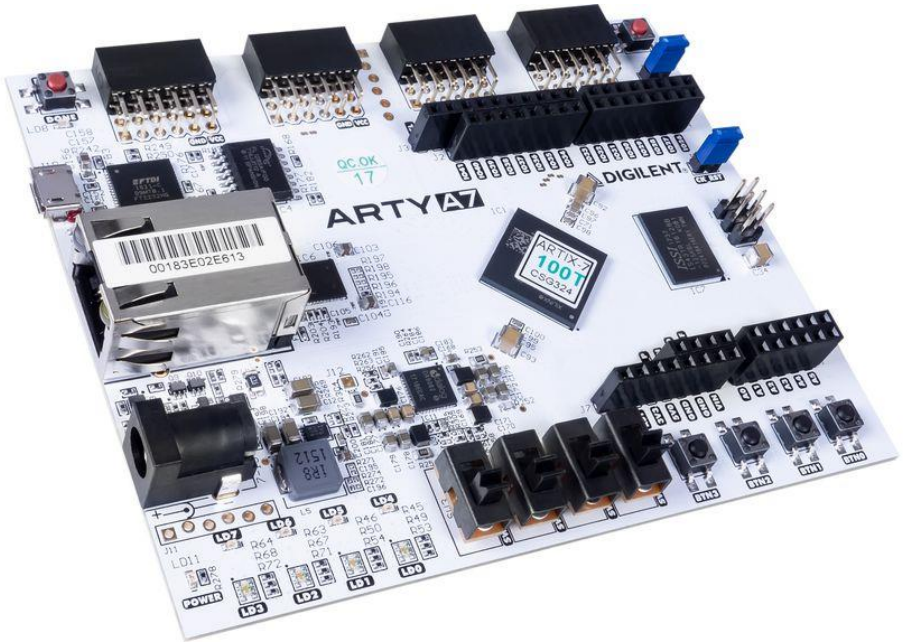
Thông qua chương 1, báo cáo đã được cung cấp thông tin khái quát về bộ tạo số ngẫu nhiên thực và chức năng của mạch tạo số ngẫu nhiên cũng như chức năng cụ thể của từng khối trong mạch. Từ bản vẽ bằng phần mềm ta kết nối các linh kiện với nhau để tạo ra mạch hoàn chỉnh.

# CHƯƠNG 2. THỰC THI TRÊN FPGA

## 2.1. Giới thiệu về FPGA

Mảng cổng lập trình được dạng trường (Field-Programmable Gate Array – FPGA) là vi mạch dùng cấu trúc mảng phần tử, được cấu hình bằng ngôn ngữ thiết kế phần cứng như VHDL hay Verilog, các khối logic có thể định cấu hình được kết nối thông qua các kết nối có thể lập trình được. Thành phần cấu tạo của FPGA gồm: các khối logic có thể tái cấu hình, các cổng I/O để giao tiếp giữa các khối logic và kiến trúc bên ngoài, kết nối trong (interconnect) để liên kết các khối logic và cổng I/O, khối ROM/RAM để lưu trữ dữ liệu. FPGA có thể được lập trình lại theo yêu cầu ứng dụng hoặc chức năng mong muốn sau khi sản xuất. FPGA được dùng để giải quyết những bài toán phức tạp, ứng dụng trong xử lí tín hiệu số, hàng không vũ trụ, mật mã học và nhiều lĩnh vực khác [12].

Arty A7 là bo mạch phát triển FPGA của hãng Digilent được xây dựng dựa trên nền tảng FPGA Artix-7 của Xilinx, sản phẩm phù hợp với cho quá trình tìm hiểu về FPGA. Nó có các cổng giao tiếp thông dụng như: UARTs, SPIs, I2Cs, được thiết kế đặc biệt để dùng làm hệ thống vi xử lý mềm MicroBlaze. Gồm 2 phiển bản là Arty-35T và Arty-100T, phiên bản được sử dụng trong báo cáo này là Arty-100T.




Hình 2.1. Kit Artix 7 100T Arty FPGA Evaluation [13]

## 2.2. Cấu hình phần cứng để kết nối mạch sinh số ngẫu nhiên với Arty

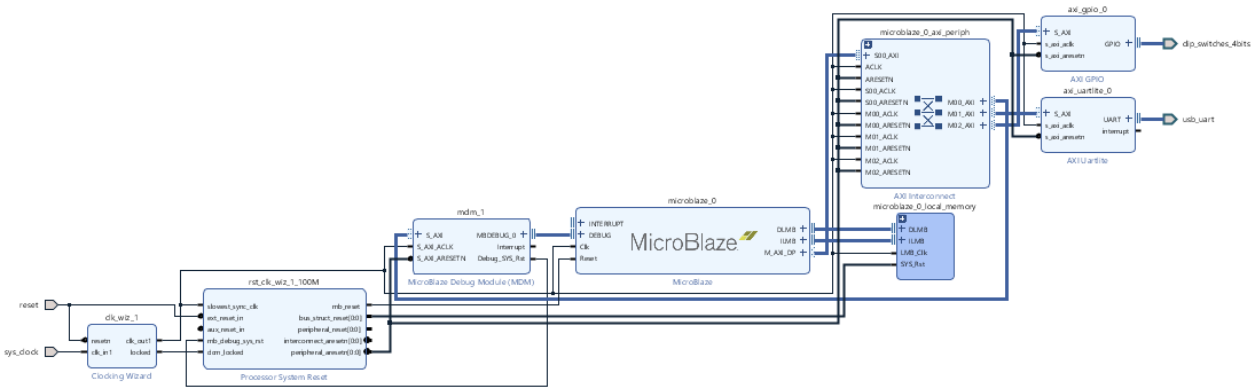
Để thực thi bộ sinh số ngẫu nhiên, trước tiên phải thiết kế và cấu hình cho FPGA. Sau đây là các bước trong quy trình cấu hình cho FPGA để nhận dữ liệu từ mạch, và hiển thị trên màn hình máy tính. Việc thiết kế được thực hiện trên phần mềm Vivado-một công cụ phần mềm của Xilinx được thực hiện theo các bước dưới đây:

1. Tạo Block Design: Block Design được tạo bằng việc thêm các IP và kết nối chúng. Bước này giúp thêm các mô-đun cần thiết cho FPGA để có thể nhận và truyền dữ liệu. Trong quá trình thiết kế Block Design có thể sử dụng công cụ Validate design để có kiểm tra thiết kế đáp ứng các yêu cầu của một hệ thống.
2. Tạo tệp wapper cho hệ thống: Tệp wapper được dùng để chuyển đổi hệ thống từ Block Design sang ngôn ngữ mô tả phần cứng. Chọn Create HDL Wrapper để tạo mô hình VHDL cấp cao nhất.
3. Tạo tệp Constraint: Tệp Constraint được dùng để ràng buộc các chân đầu ra cho hệ thống FPGA. Để ràng buộc các chân GPIO, cần chỉ định tên các chân GPIO và ánh xạ tới FPGA. Thêm dòng lệnh sau trong tệp XDC để thêm một chân GPIO cần thiết:

   set_property -dict {<Pin name>  IOSTANDARD LVCMOS33} [get_ports {GPIO name }];

4. Systhesis: Là quá trình chuyển đổi ngôn ngữ mô tả phần cứng của thiết kế thành netlist (Netlist là một danh sách các cổng và các cấu trúc logic cụ thể trong thiết kế). Bước này còn giúp điều chỉnh và tối ưu hóa thiết kế trước khi thực thi trên FPGA
5. Implementation: Bước này giúp triển khai thiết kế từ mô hình logic thành nguyên mẫu để chạy trên FPGA. Quá trình này để đảm bảo thiết kế được thực thi một cách chính xác trên FPGA.
6. Tạo Bitstream: Đây là một trong những bước quan trọng nhất trong quá trình cấu hình FPGA. Tạo một tệp bitstream để có thể nạp vào FPGA để thực hiện chức năng cụ thể đã được thiết kế.
7. Lập trình trên FPGA: Sau khi tạo tệp bitstream từ bước trước đó, cần tạo code trên phần mềm SDK. Chương trình sẽ được biên dịch và chạy trên FPGA, kết quả sẽ được hiển thị trên phần mềm.

Hình 2.2 sau đây là sơ đồ khối cấu hình phần cứng của Arty-7 100T bằng phần mềm Vivado. Sơ đồ khối phần cứng sẽ được thêm vào 2 khối UART để kết nối với máy tính và AXI GPIO để cấu hình chân đọc giá trị đầu ra của mạch tạo số ngẫu nhiên.




Hình 2.2. Cấu hình phần cứng của Arty-7 100T

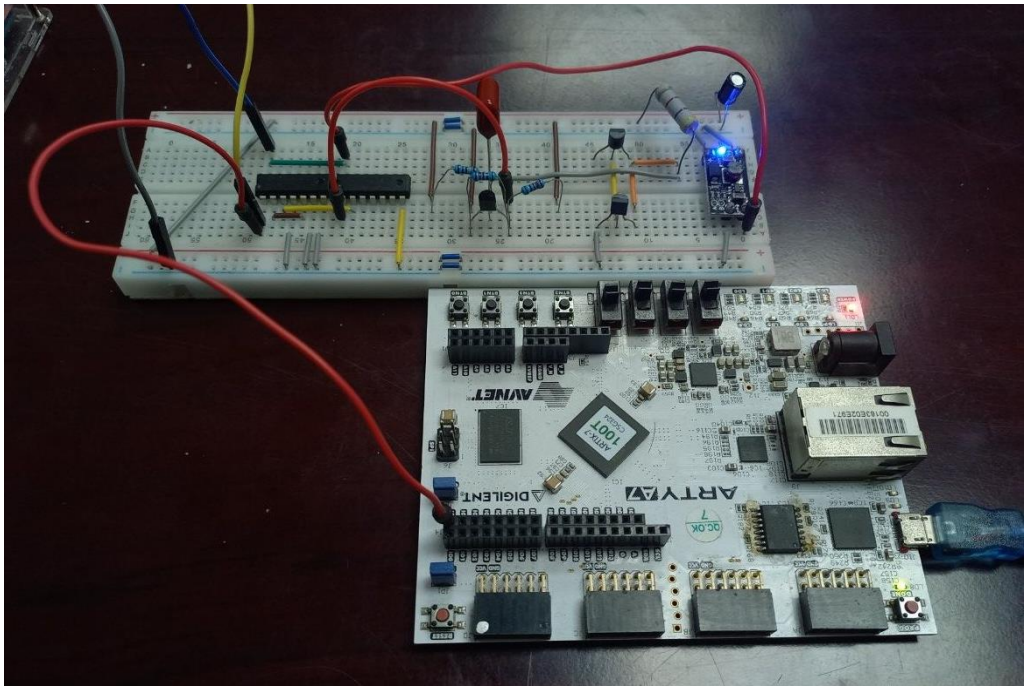
Sau khi đã cấu hình phần cứng trên Arty A7-100T, tiếp theo là quá trình kết nối và giao tiếp giữa mạch sinh số ngẫu nhiên và bo mạch Arty A7-100T.

Dữ liệu từ bộ sinh số ngẫu nhiên sẽ được truyền đến bo mạch Arty qua chân IO số 26. Để dữ liệu có thể truyền, cần thêm các ràng buộc GPIO tại tệp XDC trong quá trình cấu hình. Cần đảm bảo rằng kết nối này được thực hiện đúng cách để dữ liệu có thể truyền đến mạch Arty.

Arty A7-100T giao tiếp với máy tính thông qua cổng UART. Vì vậy, cần kết nối các chân UART của Arty với máy tính và cấu hình AXI-UART trong thiết kế. Cổng UART được sử dụng để gửi và nhận dữ liệu giữa FPGA và máy tính.

Khi Arty A7-100T được kết nối với máy tính, dữ liệu đầu ra từ mạch sinh số ngẫu nhiên có thể được truyền và hiển thị trên màn hình máy tính. Dữ liệu được biểu diễn dưới dạng các chuỗi bit 0 và 1, cần sử dụng phần mềm trên máy tính để dữ liệu hiển thị rõ ràng hơn.

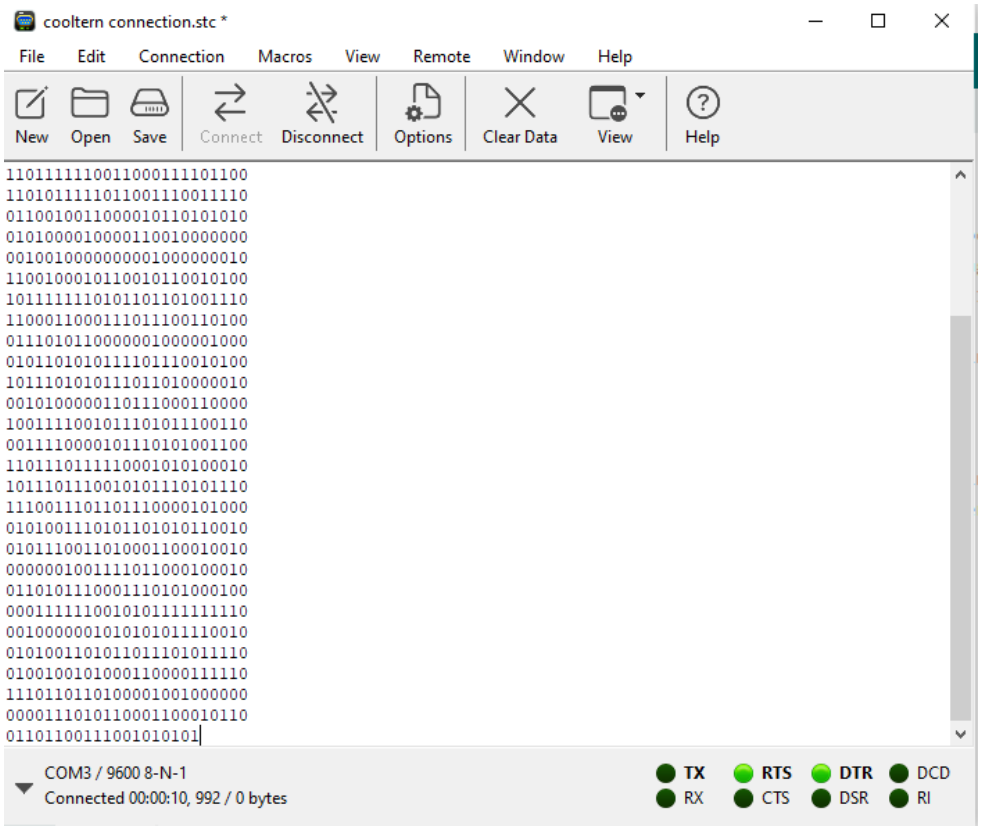
Hình 2.3 sau đây là hình ảnh ghép nối mạch sinh số ngẫu nhiên và Arty A7-100T. Sau khi ghép nối mạch, Arty và máy tính thì dữ liệu sẽ được truyền và nhận để hiển thị:




Hình 2.3. Mạch sinh số ngẫu nhiên khi kết nối với Arty

## 2.3. Dữ liệu đầu ra

Sử dụng phần mềm Coolterm để ghi dữ liệu đầu ra, dữ liệu được hiển thị trên màn hình theo các dòng, mỗi dòng 25 bit 0 và 1. Kết quả được ghi vào tệp text để phục vụ cho việc đánh giá. Qua đánh giá chủ quan, có thể thấy các dữ liệu được sinh ra có số lượng bit 0 và bit 1 đồng đều, tuy nhiên, việc đánh giá cần sử dụng các bài test để có độ chính xác cao hơn.




Hình 2.4. Kết quả chuỗi số ngẫu nhiên

## 2.4. Kết luận

Hệ thống FPGA không thể thực hiện hoạt động mà không được cấu hình trước. Để thực thi bộ sinh số ngẫu nhiên trên bo mạch Arty A7, quá trình cấu hình là bước quan trọng không thể bỏ qua. Trong quá trình này, thông tin về cách các khối logic và các thành phần kết nối của FPGA được ghi vào một tệp bitstream.

Tệp bitstream chứa các chỉ dẫn cụ thể để FPGA được cấu hình và kết nối đúng cách. Đây là quy trình quyết định cấu trúc và chức năng của FPGA để thích ứng với thiết kế cụ thể của bạn. Sau khi tạo bitstream, nó sẽ được nạp vào FPGA, các khối trên FPGA sẽ được kết nối theo thông tin trong tệp bitstream.

Vì vậy, quá trình cấu hình là bước rất quan trọng trong việc đưa hệ thống FPGA từ trạng thái chưa được cấu hình đến trạng thái có thể hoạt động theo yêu cầu thiết kế được đưa ra. Chương này đã cho thấy quy trình thiết kế và cấu hình để thực thi bộ sinh số ngẫu nhiên tích hợp vào hệ thống MicroBlaze.

# CHƯƠNG 3. ĐÁNH GIÁ KẾT QUẢ SỬ DỤNG NIST TEST

## 3.1. Giới thiệu NIST test

Bộ kiểm tra của Viện Tiêu chuẩn và Công nghệ Quốc gia Mỹ - National Institute of Standards and Technology (NIST) là gói thống kê bao gồm 16 thử nghiệm được phát triển để kiểm tra tính ngẫu nhiên của các chuỗi nhị phân (dài tùy ý) được tạo bởi các bộ tạo số ngẫu nhiên hoặc giả ngẫu nhiên bằng mật mã dựa trên phần cứng hoặc phần mềm. Các thử nghiệm này tập trung vào nhiều loại không ngẫu nhiên khác nhau có thể tồn tại theo một trình tự. Một số bài kiểm tra có thể phân tách thành nhiều bài kiểm tra phụ. Tại bảng 4.1 dưới đây là 16 bài kiểm tra. [14]

Bảng 3.1. Danh sách các bài NIST test

| STT | Tên bài kiểm tra | Mục đích |
|---|---|---|
| 1. | Frequency (Monobit) Test | Kiểm tra tỉ lệ số lượng số 1 so với số 0 trong chuỗi bit, đánh giá xem chuỗi có phân phối đều không. |
| 2. | Frequency Test within a Block | Chia chuỗi thành các khối và kiểm tra sự phân bố của số 1 và số 0 trong từng khối. |
| 3. | Runs Test | Kiểm tra số lượng chuỗi (runs) liên tục của số 1 hoặc số 0 trong chuỗi bit |
| 4. | Test for the Longest Run of Ones in a Block | Kiểm tra số lần xuất hiện của chuỗi số 1 dài nhất trong từng khối |
| 5. | Binary Matrix Rank Test | Xác định hạng của ma trận nhị phân được tạo từ chuỗi bit |
| 6. | Discrete Fourier Transform (Spectral) Test | Sử dụng phép biến đổi Fourier rời rạc để kiểm tra tính phổ của chuỗi |
| 7. | Non-overlapping Template Matching Test | Kiểm tra số lần mẫu (template) xuất hiện không chồng lấn trong chuỗi |
| 8. | Overlapping Template Matching Test | Kiểm tra số lần mẫu xuất hiện có chồng lấn trong chuỗi |
| 9. | Maurer's "Universal Statistical" Test | Sử dụng kỹ thuật thống kê để kiểm tra tính không chu kỳ của chuỗi |

| | | |
|---|---|---|
| 10. | Linear Complexity Test | Đo độ phức tạp tuyến tính của chuỗi bit |
| 11. | Serial Test | Kiểm tra sự phụ thuộc giữa các chuỗi bit liên tiếp trong chuỗi |
| 12. | Approximate Entropy Test | Đánh giá độ ngẫu nhiên của chuỗi dựa trên độ lệch của thông tin entropy xấp xỉ |
| 13. | Cumulative Sums Test (Forward) | Kiểm tra xem tổng tích luỹ của chuỗi có lớn hay nhỏ so với kỳ vọng của một chuỗi ngẫu nhiên hay không |
| 14. | Cumulative Sums Test (Backward) | |
| 15. | Random Excursions Test: | Đếm số lượng chu kỳ cụ thể trong một cuộc đi bộ ngẫu nhiên của chuỗi |
| 16. | Random Excursions Variant | Đếm số lần một trạng thái cụ thể được thăm trong một cuộc đi bộ ngẫu nhiên của chuỗi |

## 3.2. Kết quả NIST test

Bảng 3.2 sau đây là kết quả của NIST test với định dạng đầu vào bộ số ngẫu nhiên gồm 23.000.000 bit.

Bảng 3.2. Kết quả của các NIST test của 23.000.000 bit

| Bài kiểm tra | P-Value | Kết luận |
|---|---|---|
| 01. Frequency (Monobit) Test | 0.231696559 | Ngẫu nhiên |
| 02. Frequency Test within a Block | 0.47346398 | Ngẫu nhiên |
| 03. Runs Test | 0.622316246 | Ngẫu nhiên |
| 04. Test for the Longest Run of Ones in a Block | 0.195221157 | Ngẫu nhiên |
| 05. Binary Matrix Rank Test | 0.369339513 | Ngẫu nhiên |

| | | |
|---|---|---|
| 06. Discrete Fourier Transform (Spectral) Test | 0.112386576 | Ngẫu nhiên |
| 07. Non-overlapping Template Matching Test | 0.816816571 | Ngẫu nhiên |
| 08. Overlapping Template Matching Test | 0.767932496 | Ngẫu nhiên |
| 09. Maurer's "Universal Statistical" Test | 0.218412003 | Ngẫu nhiên |
| 10. Linear Complexity Test | 0.915316791 | Ngẫu nhiên |
| 11. Serial Test: | 0.512747579 | Ngẫu nhiên |
| | 0.812005813 | Ngẫu nhiên |
| 12. Approximate Entropy Test | 0.083654909 | Ngẫu nhiên |
| 13. Cumulative Sums Test (Forward) | 0.357141265 | Ngẫu nhiên |
| 14. Cumulative Sums Test (Backward) | 0.300269413 | Ngẫu nhiên |

15. Random Excursions Test:

| Trạng thái | Chi Squared | P-Value | Kết luận |
|---|---|---|---|
| -4 | 0.982896471 | 0.96393346 | Ngẫu nhiên |
| -3 | 0.832695082 | 0.974901245 | Ngẫu nhiên |
| -2 | 2.150779194 | 0.827912997 | Ngẫu nhiên |
| -1 | 2.147540984 | 0.828376245 | Ngẫu nhiên |
| 1 | 8.016393443 | 0.155334464 | Ngẫu nhiên |
| 2 | 5.38818053 | 0.370363118 | Ngẫu nhiên |
| 3 | 1.95255082 | 0.855670826 | Ngẫu nhiên |
| 4 | 4.200080568 | 0.520983661 | Ngẫu nhiên |

16. Random Excursions Variant

| Trạng thái | COUNTS | P-Value | Kết luận |
|---|---|---|---|
| -7 | 266 | 0.809136943 | Ngẫu nhiên |

| | | | |
|---|---|---|---|
| -6 | 288 | 0.607058312 | Ngẫu nhiên |
| -5 | 304 | 0.451267892 | Ngẫu nhiên |
| -4 | 302 | 0.428576985 | Ngẫu nhiên |
| -3 | 282 | 0.566378589 | Ngẫu nhiên |
| -2 | 268 | 0.681342728 | Ngẫu nhiên |
| -1 | 235 | 0.855426029 | Ngẫu nhiên |
| 1 | 217 | 0.480401901 | Ngẫu nhiên |
| 2 | 233 | 0.618521679 | Ngẫu nhiên |
| 3 | 264 | 0.365276044 | Ngẫu nhiên |
| 4 | 283 | 0.308069426 | Ngẫu nhiên |
| 5 | 293 | 0.321209231 | Ngẫu nhiên |
| 6 | 287 | 0.461904248 | Ngẫu nhiên |
| 7 | 254 | 0.880060193 | Ngẫu nhiên |
| 8 | 241 | 0.967338681 | Ngẫu nhiên |
| 9 | 256 | 0.880243464 | Ngẫu nhiên |

## 3.3. Kết luận

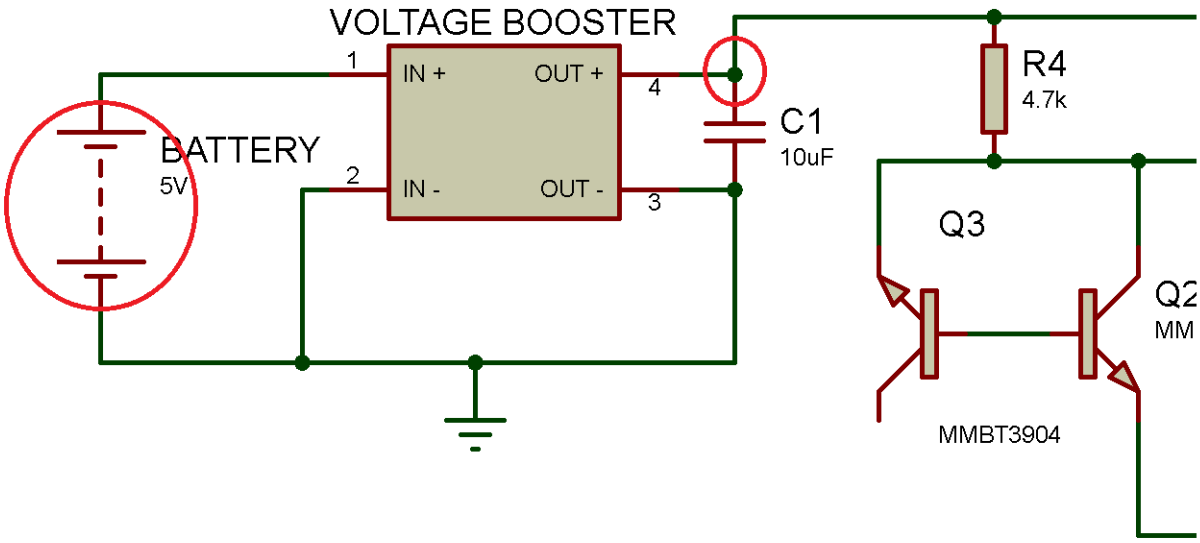
Bộ test gồm 23.000.000 bit đã vượt qua 16 bài NIST test và đưa ra kết quả là không thể dự đoán được. Với số lượng mẫu là 23.000.000 cũng đã là một bộ mẫu ở mức trung bình. Vì vậy ta có thể kết luận rằng mạch tạo số ngẫu nhiên có thể tin tưởng được.

# CHƯƠNG 4. MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG MẠCH

## 4.1. Các cách tấn công

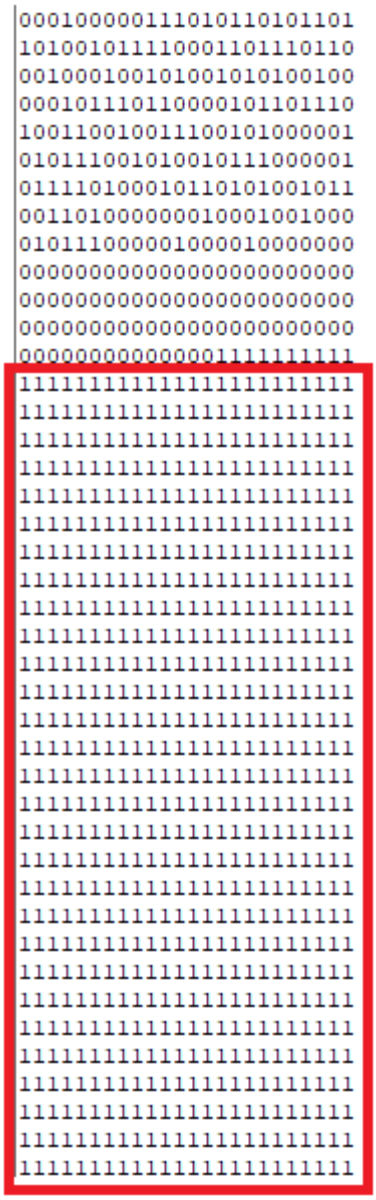
Việc tấn công vào mạch tạo số ngẫu nhiên được thực hiện nhằm để tạo ra giá trị ngẫu nhiên như phía tấn công muốn từ đó có thể dự đoán trước được kết quả mong muốn. Trong chương 4 sẽ đưa ra 2 cách để tấn công mạch tạo số ngẫu nhiên đó là thay đổi điện áp cấp nguồn và tăng nhiệt độ của nguồn entropy. [15]

### 4.1.1. Thay đổi điện áp nguồn




Hình 4.1. Vị trí tấn công điện áp nguồn

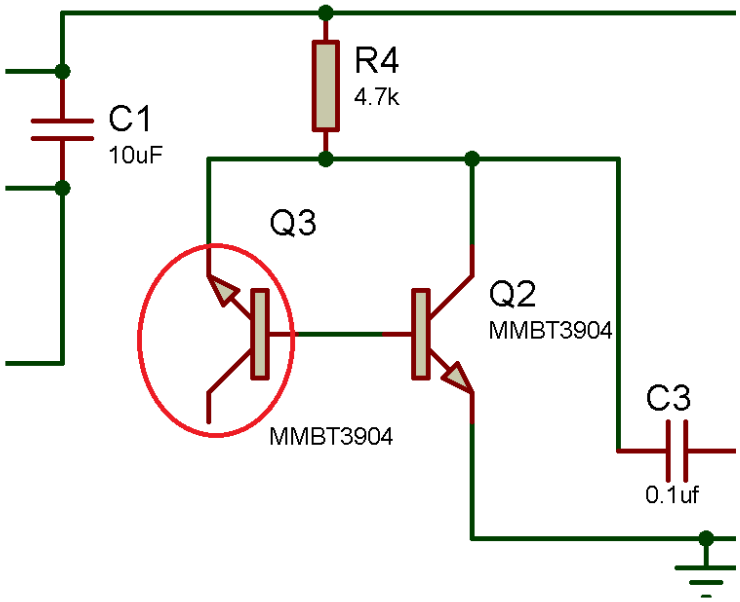
Theo hình 4.1 ta tấn công vào vị trí 1 hoặc 2 với mục đích thay đổi nguồn điện áp cấp vào Q3 hay nguồn entropy của hệ thống. Việc tấn công dựa trên nguyên lý hiệu ứng avalanche rằng nhiễu tạo bởi phân lớp p-n chỉ được tạo ra khi đáp ứng một điện áp ngược lớn điện áp đánh thủng của phân lớp đó. Việc ngắt nguồn làm cho mạch không thể tạo ra nhiễu từ đó không tạo ra được xung clock và làm cho giá trị đầu ra Q của D flip-flop dừng lại ở giá trị 0 hoặc 1 như ở hình 4.2.



Hình 4.2. Kết quả hiển thị trên màn hình sau khi ngắt nguồn cấp

### 4.1.2. Tăng nhiệt độ của nguồn entropy

Trong trường hợp này ta tấn công bằng cách dùng hơi nóng tăng nhiệt độ cho nguồn entropy transistor Q3 trong hình 4.3. Do transistor là một linh kiện điện tử do đó tồn tại một mốc nhiệt độ để transistor có thể hoạt động bình thường. Nếu tăng nhiệt độ của transistor vượt trên mốc này thì sẽ ảnh hưởng đến khả năng hoạt động của linh kiện từ đó ảnh hưởng đến nhiễu và kết quả đầu ra. Để kiểm chứng cách tấn công này thì giá trị số ngẫu nhiên trong khi thay đổi nhiệt độ sẽ được ghi lại tương tự như phương thức ở chương 3 để so sánh với mẫu thử có cùng số lượng mẫu khi hoạt động ở nhiệt độ bình thường.




Hình 4.3. Vị trí nguồn entropy để tấn công nhiệt

Bảng 4.1 bên dưới là kết quả các NIST test với 500.000 bit thử khi nguồn entropy hoạt động ở nhiệt độ bình thường.

Bảng 4.1. Kết quả của các NIST test của 500.000 bit khi nguồn entropy hoạt động ở nhiệt độ bình thường

| Bài kiểm tra | P-Value | Kết luận |
|---|---|---|
| 01. Frequency (Monobit) Test | 0.048687609 | Ngẫu nhiên |
| 02. Frequency Test within a Block | 0.016840464 | Ngẫu nhiên |
| 03. Runs Test | 0.665579607 | Ngẫu nhiên |
| 04. Test for the Longest Run of Ones in a Block | 0.338095928 | Ngẫu nhiên |
| 05. Binary Matrix Rank Test | 0.550771115 | Ngẫu nhiên |
| 06. Discrete Fourier Transform (Spectral) Test | 0.012599748 | Ngẫu nhiên |
| 07. Non-overlapping Template Matching Test | 0.01024574 | Ngẫu nhiên |
| 08. Overlapping Template Matching Test | 0.58036056 | Ngẫu nhiên |

| | | |
|---|---|---|
| 09. Maurer's "Universal Statistical" Test | 0.291774839 | Ngẫu nhiên |
| 10. Linear Complexity Test | 0.901056092 | Ngẫu nhiên |
| 11. Serial Test: | 0.142174819 | Ngẫu nhiên |
| | 0.054794836 | Ngẫu nhiên |
| 12. Approximate Entropy Test | 0.116742395 | Ngẫu nhiên |
| 13. Cumulative Sums Test (Forward) | 0.015707594 | Ngẫu nhiên |
| 14. Cumulative Sums Test (Backward) | 0.032460313 | Ngẫu nhiên |

15. Random Excursions Test:

| Trạng thái | Chi Squared | P-Value | Kết luận |
|---|---|---|---|
| -4 | 13.40917864 | 0.019831654 | Ngẫu nhiên |
| -3 | 1.740423529 | 0.883768242 | Ngẫu nhiên |
| -2 | 1.352860486 | 0.929399542 | Ngẫu nhiên |
| -1 | 1.620915033 | 0.898708596 | Ngẫu nhiên |
| 1 | 0.183006536 | 0.999285976 | Ngẫu nhiên |
| 2 | 4.343419672 | 0.501102525 | Ngẫu nhiên |
| 3 | 11.00828235 | 0.051216016 | Ngẫu nhiên |
| 4 | 3.124868995 | 0.680741659 | Ngẫu nhiên |

16. Random Excursions Variant

| Trạng thái | COUNTS | P-Value | Kết luận |
|---|---|---|---|
| -7 | 303 | 0.97653619 | Ngẫu nhiên |
| -6 | 349 | 0.653580213 | Ngẫu nhiên |
| -5 | 359 | 0.552382164 | Ngẫu nhiên |
| -4 | 335 | 0.723752353 | Ngẫu nhiên |
| -3 | 347 | 0.580645386 | Ngẫu nhiên |
| -2 | 341 | 0.592829111 | Ngẫu nhiên |

| | | | |
|---|---|---|---|
| -1 | 330 | 0.664389443 | Ngẫu nhiên |
| 1 | 339 | 0.44120852 | Ngẫu nhiên |
| 2 | 323 | 0.491966615 | Ngẫu nhiên |
| 3 | 308 | 0.935565055 | Ngẫu nhiên |
| 4 | 311 | 0.907105709 | Ngẫu nhiên |
| 5 | 340 | 0.538794938 | Ngẫu nhiên |
| 6 | 386 | 0.221607558 | Ngẫu nhiên |
| 7 | 380 | 0.318720388 | Ngẫu nhiên |
| 8 | 351 | 0.583379988 | Ngẫu nhiên |
| 9 | 359 | 0.552382164 | Ngẫu nhiên |

Bảng 4.2 bên dưới là kết quả bài kiểm tra NIST test với cùng số mẫu thử khi tác động nhiệt độ vào nguồn entropy

Bảng 4.2. Kết quả của các NIST test của 500.000 bit khi nguồn entropy bị tác động bởi nhiệt độ

| Bài kiểm tra | P-Value | Kết luận |
|---|---|---|
| 01. Frequency (Monobit) Test | 0 | Không ngẫu nhiên |
| 02. Frequency Test within a Block | 0 | Không ngẫu nhiên |
| 03. Runs Test | 0 | Không ngẫu nhiên |
| 04. Test for the Longest Run of Ones in a Block | 1.78E-20 | Không ngẫu nhiên |
| 05. Binary Matrix Rank Test | 0.001218534 | Không ngẫu nhiên |
| 06. Discrete Fourier Transform (Spectral) Test | 8.10E-86 | Không ngẫu nhiên |
| 07. Non-overlapping Template Matching Test | 0 | Không ngẫu nhiên |
| 08. Overlapping Template Matching Test | 0.575842126 | Ngẫu nhiên |

| | | |
|---|---|---|
| 09. Maurer's "Universal Statistical" Test | 0 | Không ngẫu nhiên |
| 10. Linear Complexity Test | 0.017113809 | Ngẫu nhiên |
| 11. Serial Test: | 0 | Không ngẫu nhiên |
| | 0 | Không ngẫu nhiên |
| 12. Approximate Entropy Test | 0 | Không ngẫu nhiên |
| 13. Cumulative Sums Test (Forward) | 0 | Không ngẫu nhiên |
| 14. Cumulative Sums Test (Backward) | 0 | Không ngẫu nhiên |

| 15. Random Excursions Test: | | | |
|---|---|---|---|
| Trạng thái | Chi Squared | P-Value | Kết luận |
| -4 | 13.40917864 | 0.019831654 | Ngẫu nhiên |
| -3 | 1.740423529 | 0.883768242 | Ngẫu nhiên |
| -2 | 1.352860486 | 0.929399542 | Ngẫu nhiên |
| -1 | 1.620915033 | 0.898708596 | Ngẫu nhiên |
| 1 | 0.183006536 | 0.999285976 | Ngẫu nhiên |
| 2 | 4.343419672 | 0.501102525 | Ngẫu nhiên |
| 3 | 11.00828235 | 0.051216016 | Ngẫu nhiên |
| 4 | 3.124868995 | 0.680741659 | Ngẫu nhiên |

| 16. Random Excursions Variant | | | |
|---|---|---|---|
| Trạng thái | COUNTS | P-Value | Kết luận |
| -7 | 22 | 0.72413254 | Ngẫu nhiên |
| -6 | 23 | 0.730527924 | Ngẫu nhiên |
| -5 | 21 | 0.66193925 | Ngẫu nhiên |
| -4 | 20 | 0.608727509 | Ngẫu nhiên |
| -3 | 13 | 0.395951025 | Ngẫu nhiên |
| -2 | 14 | 0.359300654 | Ngẫu nhiên |

| | | | |
|---|---|---|---|
| -1 | 20 | 0.447699072 | Ngẫu nhiên |
| 1 | 21 | 0.362726506 | Ngẫu nhiên |
| 2 | 27 | 0.395951025 | Ngẫu nhiên |
| 3 | 38 | 0.627625805 | Ngẫu nhiên |
| 4 | 39 | 0.726286149 | Ngẫu nhiên |
| 5 | 40 | 0.74488162 | Ngẫu nhiên |
| 6 | 57 | 0.291789863 | Ngẫu nhiên |
| 7 | 77 | 0.082180515 | Ngẫu nhiên |
| 8 | 97 | 0.02125027 | Ngẫu nhiên |
| 9 | 110 | 0.010583547 | Ngẫu nhiên |

Như có thể thấy ở 2 bảng trên, sau khi tăng nhiệt độ của nguồn entropy lên thì giá trị ngẫu nhiên thu được trở nên dễ đoán hơn dựa trên kết quả của NIST test. Vì vậy ta có thể kết luận rằng tăng nhiệt độ nguồn entropy là một trong những cách hiệu quả để tấn công mạch tạo số ngẫu nhiên.

## 4.2. Kết luận

Thông qua chương 4 ta biết thêm được 2 cách tấn công vào mạch nhằm với mục đích vô hiệu hóa mạch tạo số ngẫu nhiên hoặc thao túng mạch để tạo ra giá trị mong muốn. Tuy nhiên, 2 phương pháp trên yêu cầu bên tấn công phải tương tác vật lý trực tiếp với mạch để có thể thay đổi. Do đó một trong những phương pháp để ngăn chặn kiểu tấn công này là bảo vệ mạch và nguồn cấp mạch tai những vị trí mà bên tấn công không thể tiếp xúc được và ít bị ảnh hưởng bởi yếu tố môi trường.

## TÀI LIỆU THAM KHẢO


[1] M. S. ˇ. c. a. Ç. K. Koç, *True Random Number Generators,* 2014.

[2] V. F. a. M. Drutarovsk´y, *True Random Number Generator Embedded in Reconfigurable Hardware,* 2002.

[3] P. L'Ecuyer, *Random Number Generation,* 2012.

[4] jongrover, "github.com," 2022. [Online]. Available: https://github.com/jongrover/true-random?tab=readme-ov-file.

[5] B. Subedi, "how2electronics," Boost Converter: Basics, Working, Design & Application, [Online]. Available: https://how2electronics.com/boost-converter-basics-working-design-application/.

[6] D. Rom˜ao, *Extremely Secure Communication,* 2015.

[7] Madhu-SUDAN-GUPTA, *Noise in Avalanche Transit-Time Devices,* 1971.

[8] "altervista," Random Sequence Generator based on Avalanche Noise , [Online]. Available: http://holdenc.altervista.org/avalanche/.

[9] N. C. Braga, "incbtech.com," [Online]. Available: https://www.incbtech.com/articles/17-paranormal-electronic/333-white-noise-generator-art078.html.

[10] ONSEMI, "alldatasheet.com," 2006. [Online]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/12160/ONSEMI/MC74HC14A.html.

[11] "electronics-tutorial.net," Toggle Flip-flop, [Online]. Available: https://www.electronics-tutorial.net/sequential-logic-circuits/toggle-flip-flop/.

[12] A. Magyari, "mdpi," Review of State-of-the-Art FPGA Applications in IoT Networks, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7496.

[13] "reichelt," [Online]. Available: https://www.reichelt.com/it/en/arty-a7-100t-artix-7-fpga-development-board-digil-410-319-1-p285629.html?&nbc=1.

[14] J. J. M. S. E. L. M. M. V. D. A. J. S. AndrewRukhin, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, 2010.

[15] A. P. V. R. I. V. Yrjo Koyen, *Attacking Hardware Random Number Generators in a Multi-Tenant Scenario*, 2020.

[16] "linhkien3m.vn," [Online]. Available: https://linhkien3m.vn/module-boost-dcdc-ghim-dien-ap-ra-5v8v9v12v-chuyen-dung-cho-pin-lithium-p27681827.html.

[17] N. Semiconductors, "alldatasheet.com," 74HC14 Datasheet (PDF), 1997. [Online]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/112419/PHILIPS/74HC14.html.

[18] T. INSTRUMENTS, CD4013B CMOS Dual D-Type Flip-Flop, 1998. [Online]. Available: https://www.ti.com/lit/ds/symlink/cd4013b.pdf.

[19] theenggprojects, "rs-online.com," Basics of 2N3904, 2018. [Online]. Available: https://www.rs-online.com/designspark/basics-of-2n3904.

[20] "pchcables.com," 1/4W 1% Metal Film Resistor 4.7k ohm, [Online]. Available: https://www.pchcables.com/83-1033.html.

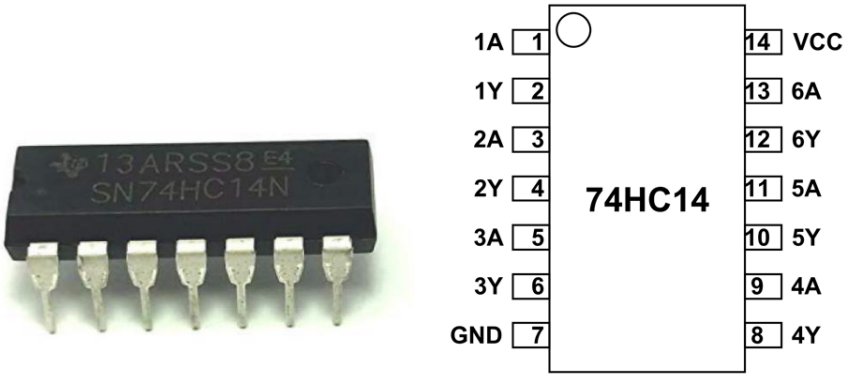
[21] "vietnic.vn," TỤ 0.1UF - 50V, [Online]. Available: https://www.vietnic.vn/tu-0-1uf-50v.

[22] "vietnic.v," TỤ 10UF - 50V, [Online]. Available: https://www.vietnic.vn/tu-hoa-10uf-50v.

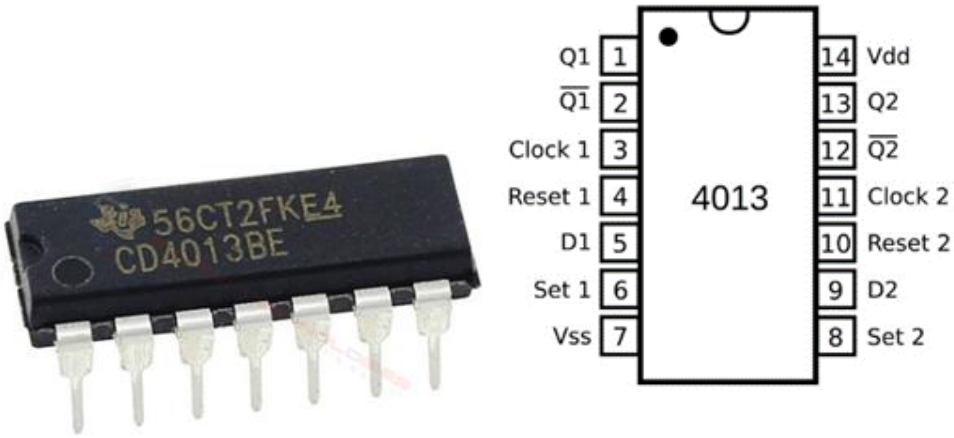
[23] "core-electronics.com.au," How to Use Breadboards, [Online]. Available: https://core-electronics.com.au/guides/how-to-use-breadboards/.

# PHỤ LỤC A

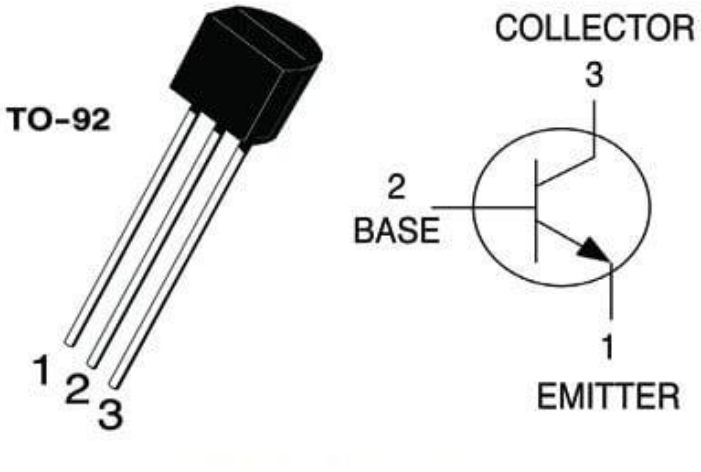

Hình 1. Module tăng điện áp đầu vào từ 5V lên 12V [16]



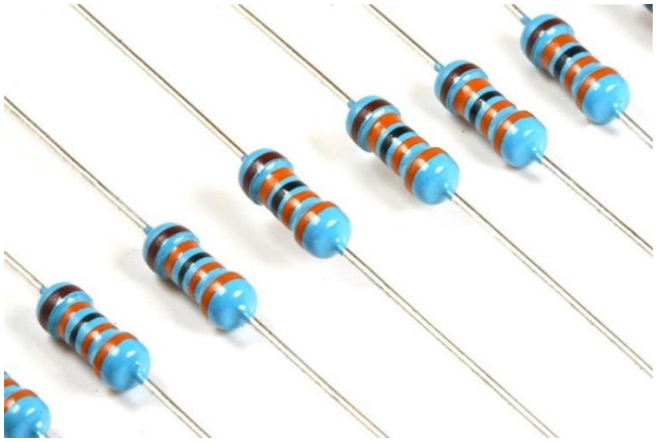

Hình 2. Hex-Schmitt Trigger 74HC14 [17]




Hình 3. Mạch tích hợp CD4013BE (D-flip flop) [18]



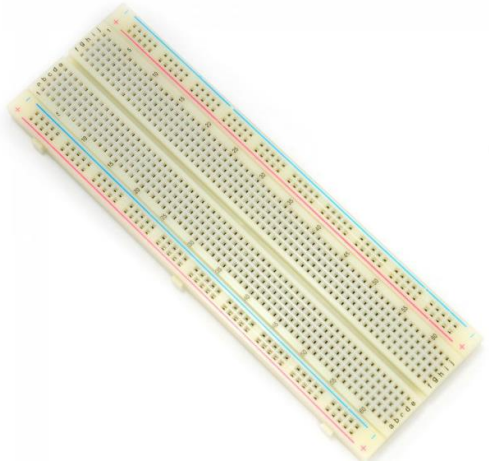

Hình 4. Transistor loại NPN 2N 3904 [19]



Hình 5. Điện trở 4.7K Ohm [20]


Hình 6. Tụ điện 0.1 microFarad [21]


Hình 7. Tụ điện 10 microFarad [22]



Hình 8. Breadboard [23]

## PHỤ LỤC B

Bảng 1. Thông số cơ bản của module tăng điện áp [16]

| | |
|---|---|
| Kích thước | 22x11mm |
| Khối lượng | 10g |
| Đầu ra (đầu vào 3.7 V) | 12V 0.3A |
| Đầu ra (đầu vào 5V) | 12V 0.3A |

Bảng 2. Mô tả chân của IC 74HC14 [17]

| Pin | Mô tả |
|---|---|
| A1, A2, A3, A4, A5, A6 | Chân đầu vào |
| Y1, Y2, Y3, Y4, Y5, Y6 | Chân đầu ra |
| Vcc | Chân cấp nguồn |
| GND | Chân nối đất |

Bảng 3. Mô tả chân của CD4013BE [18]

| Pin | Mô tả |
|---|---|
| Q | Chân tín hiệu ra |
| $\overline{Q}$ | Chân tín hiệu đảo của Q |
| Clock | Lối vào xung clock |
| D | Lối vào dữ liệu (Data). |
| Set | Thiết lập đầu ra là 1 |
| Reset | Thiết lập đầu ra là 0 |
| Vss | Chân nối đất |
| Vdd | Điện áp nguồn |

# PHỤ LỤC C

Code lập trình cho FPGA https://github.com/jake-newbie/True-random-number-generator-on-FPGA