|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  Ảnh có chứa văn bản, ký hiệu, thực phẩm  Mô tả được tạo tự động**TRƯỜNG ĐIỆN – ĐIỆN TỬ**  **BÁO CÁO BÀI TẬP LỚN**  **HỆ THỐNG NHÚNG**  **VÀ THIẾT KẾ GIAO TIẾP NHÚNG**  **Đề tài: Hệ thống máy rửa chén**  **Nhóm sinh viên thực hiện:**   |  |  |  | | --- | --- | --- | | **Tên sinh viên** | **MSSV** | **Mã lớp** | | Nguyễn Thành Trung | 20182837 | 137337 | | Nguyễn Thanh Thủy | 20182815 | 137337 | | Ngô Văn Tuấn  Trịnh Xuân Việt | 20182859  20193194 | 137337  137337 |   **Giảng viên hướng dẫn:** TS. Phạm Văn Tiến  Hà Nội, 12-2022 |

# LỜI NÓI ĐẦU

Thời gian gần đây, các Hệ thống nhúng – Thời gian thực được quan tâm nhiều hơn ở Việt Nam, và trên thế giới thì các hệ thống này đã và đang được phát triển mạnh mẽ và là xu hướng thịnh hành ở các nước Công nghiệp vì những lợi ích to lớn, thiết thực mà nó mang lại. Chương trình học môn Hệ thống nhúng và thiết kế giao tiếp nhúng là một phần quan trọng giúp hiểu rõ quy trình thiết kế, đánh giá hệ thống nhúng. Trong báo cáo này, nhóm chúng em triển khai thiết kế và mô phỏng hệ thống nhúng cho máy rửa bát tự động. Thiết kế được triển khai bằng ngôn ngữ mô tả phần cứng Verilog, System Verilog và ngôn ngữ C, mô phỏng kiểm thử trên phần mềm QuestaSim và tổng hợp trên phần mềm Vivado. Cho ra kết quả hoạt động đúng với yêu cầu bài toán. Chúng em sẽ đi trình bày cụ thể những gì chúng em đã làm được thông qua 6 chương sau:

**Chương 1: Giới thiệu chung (Introduction)**

**Chương 2: Đặc tả hệ thống (System Specification)**

**Chương 3: Thiết kế hệ thống (System Design)**

**Chương 4: Triển khai trên FPGA (Implement FPGA)**

**Chương 5: Kiểm thử (Verification)**

**Chương 6: Kết luận**

Nhóm chúng em xin chân thành cảm ơn thầy Phạm Văn Tiến đã tận tâm hướng dẫn nhóm chúng em trong quá trình thực hiện bài tập lớn cũng như hoàn hiện báo cáo này.

# MỤC LỤC

[LỜI NÓI ĐẦU ii](#_Toc124976522)

[MỤC LỤC iii](#_Toc124976523)

[DANH MỤC HÌNH VẼ i](#_Toc124976524)

[DANH MỤC BẢNG BIỂU iii](#_Toc124976525)

[CHƯƠNG 1. GIỚI THIỆU CHUNG (INTRODUCTION) 1](#_Toc124976526)

[1.1 Giới thiệu chung 1](#_Toc124976527)

[1.2 Hệ thống thời gian thực (Real-time operating system) 1](#_Toc124976528)

[1.3 Hệ thống máy rửa bát tự động 2](#_Toc124976529)

[CHƯƠNG 2. MÔ HÌNH HÓA HỆ THỐNG 5](#_Toc124976530)

[2.1 Mô hình hóa hệ thống sử dụng UML 5](#_Toc124976531)

[2.1.1 Use case diagram 5](#_Toc124976532)

[2.1.2 Class Diagram 6](#_Toc124976533)

[2.1.3 State Diagram 7](#_Toc124976534)

[2.1.4 Sequence Diagram 7](#_Toc124976535)

[2.2 Mô hình hóa hệ thống sử dụng SystemC 12](#_Toc124976536)

[2.3 Mô hình bằng FSM 14](#_Toc124976537)

[2.3.1 Tổng quan về FSM 14](#_Toc124976538)

[2.3.2 Tổng quan về HDL 17](#_Toc124976539)

[2.3.3 Thiết kế máy rửa bát 17](#_Toc124976540)

[2.3.4 Chi tiết về bộ điều khiển máy rửa bát 17](#_Toc124976541)

[2.3.5 Quy trình thiết kế 18](#_Toc124976542)

[2.3.6 Thiết kế cho phần phân loại chén đĩa 18](#_Toc124976543)

[2.3.7 Thiết kế cho phần rửa 21](#_Toc124976544)

[2.3.8 Mô phỏng 24](#_Toc124976545)

[2.4 Kết luận 26](#_Toc124976546)

[CHƯƠNG 3. THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN) 27](#_Toc124976547)

[3.1 Thiết kế phần cứng 27](#_Toc124976548)

[3.1.1 Mô tả chức năng 27](#_Toc124976549)

[3.1.2 Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath) 27](#_Toc124976550)

[3.1.3 Thiết kế khối register\_block cho system\_controller 29](#_Toc124976551)

[3.1.4 Thiết kế RTL 30](#_Toc124976552)

[3.2 Thiết kế phần mềm 30](#_Toc124976553)

[3.2.1 Mục tiêu 30](#_Toc124976554)

[3.2.2 Lý do lựa chọn phần mềm 30](#_Toc124976555)

[3.2.3 Thiết kế 32](#_Toc124976556)

[3.2.4 Hướng triển khai 34](#_Toc124976557)

[3.2.5 Sơ đồ thuật toán 34](#_Toc124976558)

[CHƯƠNG 4. TRIỂN KHAI TRÊN FPGA (IMPLEMENT FPGA) 36](#_Toc124976559)

[4.1 Tổng quan về triển khai hệ thống trên FPGA 36](#_Toc124976560)

[4.2 Các IP dùng trong hệ thống 37](#_Toc124976561)

[4.2.1 Smart system controller (system\_controller) 37](#_Toc124976562)

[4.2.2 Oled controller (oled\_controller) 39](#_Toc124976563)

[4.2.3 DHT11 controller (dht11\_controller) 40](#_Toc124976564)

[4.3 Tổng hợp và triển khai trên phần mềm Vivado 42](#_Toc124976565)

[4.3.1 Tạo Block design 42](#_Toc124976566)

[4.3.2 Tạo constraint file 42](#_Toc124976567)

[4.3.3 Tổng hợp (systhesis) 43](#_Toc124976568)

[4.3.4 Triển khai (implement) 43](#_Toc124976569)

[4.3.5 Kết quả 44](#_Toc124976570)

[CHƯƠNG 5. KIỂM THỬ (VERIFICATION) 46](#_Toc124976571)

[5.1 Kế hoạch kiểm thử 46](#_Toc124976572)

[5.1.1 Kiểm thử thiết kế phần cứng máy rửa bát sử dụng QuestaSim 46](#_Toc124976573)

[5.2 Mô phỏng trên proteus 47](#_Toc124976574)

[CHƯƠNG 6. KẾT LUẬN 48](#_Toc124976575)

[TÀI LIỆU THAM KHẢO 49](#_Toc124976576)

# DANH MỤC HÌNH VẼ

[Hình 2.1 Use case diagram của hệ thống 5](#_Toc124976622)

[Hình 2.3 State diagram của hệ thống 7](#_Toc124976623)

[Hình 2.4 Sequence diagram của hệ thống xả 8](#_Toc124976624)

[Hình 2.5 Sequence diagram của hệ thống sấy khô 9](#_Toc124976625)

[Hình 2.6 Sequence diagram của hệ thống thải nước tự động 10](#_Toc124976626)

[Hình 2.7 Sequence diagram của hệ thống làm ấm tự động 11](#_Toc124976627)

[Hình 2.9 Sơ đồ hệ thống rửa bát tự động sử dụng SystemC 12](#_Toc124976628)

[Hình 3.1 Sơ đồ ASMD 28](#_Toc124976629)

[Hình 3.2 Sơ đồ khối Control Unit và Datapath 29](#_Toc124976630)

[Hình 3.3 Các thanh ghi trong system\_controller 29](#_Toc124976631)

[Hình 3.4 Sơ đồ phần mềm hệ thống 33](#_Toc124976632)

[Hình 3.5 Sơ đồ thuật toán 34](#_Toc124976633)

[Hình 3.6 Mô hình giao tiếp giữa vi xử lý và các module 35](#_Toc124976634)

[Hình 4.1. Sơ đồ khối của hệ thống 36](#_Toc124976635)

[Hình 4.2 Sơ đồ khối của smart system controller 38](#_Toc124976636)

[Hình 4.3 Sơ đồ khối của oled\_controller 39](#_Toc124976637)

[Hình 4.4 Sơ đồ khối của DHT11 controller 41](#_Toc124976638)

[Hình 4.5 Sơ đồ thời gian quá trình gửi tín hiệu điều khiển từ master và phản hồi tín hiệu từ slave của cảm biến DHT11 42](#_Toc124976639)

[Hình 4.7 Block design của hệ thống 42](#_Toc124976640)

[Hình 4.8 Báo cáo về timing 44](#_Toc124976641)

[Hình 4.9 Báo cáo về công suất của thiết kế 44](#_Toc124976642)

[Hình 4.11 Kết quả trên kit FPGA 44](#_Toc124976643)

[Hình 5.1 Kết quả mô phỏng khối clock\_generator 46](#_Toc124976644)

[Hình 5.2 Kết quả mô phỏng khối controller 46](#_Toc124976645)

[Hình 5.3 Kết quả mô phỏng khối core 47](#_Toc124976646)

# BẢNG PHÂN CÔNG VIỆC

|  |  |
| --- | --- |
| Tên | Công việc được giao |
| Nguyễn Thành Trung | Xác lập chỉ tiêu kĩ thuật, thiết kế phần cứng, phần mềm, kiểm thử phần cứng và phần mềm trên vivado và questasim, mô hình hóa bằng UML, viết báo cáo, thuyết trình, viết code Verilog và C |
| Trịnh Xuân Việt | Mô hình hóa FSM,code Verilog cho FSM, thuyết trình, viết báo cáo |
| Nguyễn Thanh Thủy | Mô hình hóa bằng systemC,thiết kế phần mềm, viết báo cáo, thuyết trình |
| Ngô Văn Tuấn | Mô phỏng trên proteus, viết báo cáo, thuyết trình, thiết kế phần mềm |

# DANH MỤC BẢNG BIỂU

[Bảng 1.1 Bảng thông số kỹ thuật 2](#_Toc107948081)

[Bảng 4.1 Ràng buộc các tín hiệu của system\_controller 23](#_Toc107948082)

[Bảng 4.2 Tổ chức thanh ghi trong oled\_controller 25](#_Toc107948083)

[Bảng 4.3 Ràng buộc các tín hiệu của oled\_controller 25](#_Toc107948084)

[Bảng 4.4 Tổ chức thanh ghi trong dht11\_controller 26](#_Toc107948085)

[Bảng 4.5 Ràng buộc các tín hiệu của dht11\_controller 26](#_Toc107948086)

[Bảng 4.6 Số lượng LUT và Register của thiết kế 29](#_Toc107948087)

# GIỚI THIỆU CHUNG (INTRODUCTION)

Chương này giới thiệu khái quát và các kiến thức cơ bản về hệ thống nhúng bài toán máy rửa bát tự động.

## Giới thiệu chung

Hệ thống nhúng (Embedded Systems) là sự kết hợp của phần cứng và phần mềm máy tính, và có thể bổ sung phần cơ khí hoặc một số bộ phận khác, được thiết kế để thực hiện một chức năng cụ thể. Điều này trái ngược hoàn toàn với máy tính cá nhân như Laptop, Computer, … Nó cũng bao gồm phần cứng, phần mềm các thành phần cơ khí. Tuy nhiên, máy tính cá nhân không được thiết kế để thực hiện một chức năng cụ thể. Đúng hơn là nó thực hiện nhiều công việc khác nhau, hay có thể sử dụng thuật ngữ máy tính đa năng để phân biệt mới máy tính hệ nhúng.

Thông thường, một hệ thống nhúng là một thành phần trong một số hệ thống lớn hơn. Như bài toán ta đang triển khai Điều khiển máy rửa bát tự động, một hệ thống nhúng điều khiển xả nước, rửa và làm sạch tự động.

## Hệ thống thời gian thực (Real-time operating system)

Trong các bài toán điều khiển chúng ta hay bắt gặp các thuật ngữ “Thời gian thực”. Thời gian thực không phải là thời gian phản ánh một cách trung thực, chính xác thời gian hay yêu cầu hệ thống phải trùng với thời gian thực tế.

Hệ thống thời gian thực được hiểu là các hoạt động của hệ thống phải thỏa mãn về tính tiền định. Tính tiền định là hành vi của hệ thống phải được thực hiên trong một khung thời gian cho trước hoàn toàn xác định, khung thời gian này được quyết định bởi đặc điểm và yêu cầu của hệ thống.

Thực tế cho thấy rằng hầu hết các hệ thống nhúng là các hệ thống thời gian thực và ngược lại các hệ thống thời gian thực là hệ thống nhúng.

## Hệ thống máy rửa bát tự động

Không như cách rửa chén bằng tay thông thường, máy rửa bát là một thiết bị hoạt động tự động giúp làm sạch chén bát, ly tách dựa vào lực phun của vòi nước. Một hỗn hợp nước và chất tẩy rửa bắn ra với áp lực cao làm sạch các vết bẩn bám trên vật dụng. Các kệ rổ ở bên trong sẽ xoay để thay đổi vị trí, từ đó làm tăng khả năng tiếp xúc của vật dụng với nước. Sau quy trình phun xả làm sạch, máy sẽ hấp khô dụng cụ bếp bằng luồng không khí nóng.

Hệ thống của một bồn cầu tự động bao gồm:

- Nút điều khiển và màn hình

- Khóa

- Giá đỡ trên

- Gasket cánh xửa

- Hệ thống tay phun trên

- Hệ thống tay phun dưới

- Van nổi

- Ngăn chứa nước rửa bát

- Chân đến

- Van nước vào

- Ống thoát nước

- Lớp vỏ cách nhiệt, điện

- Nguồn điện

Diagram

Description automatically generated

Các thông số kĩ thuật được mô tả qua bảng sau:

Bảng . Bảng thông số kỹ thuật

|  |  |
| --- | --- |
| **Item** | **Parameter** |
| Processor | Xilinx Zynq®-7000 All Programmable SoC |
| Temperature threshold (oC) | 20 |
| Temperature sensor | DHT11 |
| Detect defecate/urinate threshold (sec) | 60 |
| Spray time | 60 sec |
| Drying time | 80 sec |
| Discharge time | 30 sec |
| Power Supply | AC 220V |
| Power of Processor | 100MHz |
| Frequency of Processor | 1.679 W |

Yêu cầu về công nghệ:

* Dễ điều khiển
* An toàn tuyệt đối cho người và thiết bị
* Xử lí các tình huống phát sinh chuẩn xác
* Tiết kiệm điện năng
* Dễ dàng nâng cấp chỉnh sửa, bảo trì
* Đảm bảo tính thời gian thực

# MÔ HÌNH HÓA HỆ THỐNG

Chương này mô tả tổng quan về hệ thống rửa bát tự động mà nhóm thực hiện.

## Mô hình hóa hệ thống sử dụng UML

UML (Unified Modeling Language) là một ngôn ngữ được sử dụng để mô tả các đặc điểm của hệ thống. Dựa vào chỉ tiêu kĩ thuật đã được đề cập trong chương 1, nhóm thực hiện thiết kế hệ thống rửa bát tự động sử dụng UML.

### Use case diagram

Tất cả các phần của hệ thống tương tác với con người hoặc các tác nhân tự động sử dụng hệ thống cho một số mục đích và cả con người và tác nhân đều mong đợi hệ thống hoạt động theo những cách có thể dự đoán được. Trong UML, use case diagram được sử dụng để mô hình hóa các hành vi của một hệ thống hoặc một phần của hệ thống.

Diagram

Description automatically generated

Hình . Use case diagram của hệ thống

Dựa theo các yêu cầu kĩ thuật đã được nêu trước đó, use case diagram của hệ thống rửa bát tự động được thể hiện ở hình trên.

Tác nhân trong hệ thống máy rửa bát tự động là người sử dụng, đây là vai trò của con người khi tương tác với hệ thống. Người sử dụng tương tác với hệ thống. Do đó, biểu đồ use case cho thấy tác nhân có mối quan hệ với các chức năng của hệ thống: Đóng/mở, làm ấm nước, tự động rửa , sấy khô, xả nước, tự chiếu sáng. Người dùng hoàn toàn có thể tắt hoặc bật chức năng tự động nhằm mục đích vệ sinh hay bảo trì hệ thống.

### Class Diagram

Biểu đồ lớp (Class Diagram) sẽ làm rõ các lớp, giao diện và mối quan hệ giữa chúng. Biểu đồ lớp liên quan đến mô tả hệ thống một cách tổng quan bao gồm các thuộc tính,hoạt động trong một lớp. Các thuộc tính và hoạt động có thể có của một hệ thống rửa bát tự động được thể hiện trên hình dưới đây.

Hệ thống rửa bát tự động gồm hai phần chính là control và Software. control là thành phần chính của hệ thống, có nhiệm vụ nhận tín hiệu từ các đầu vào như nút bấm,… đồng thời tương tác với phần mềm để điều khiển . Phần mềm xử lý các trạng thái, nhận tín hiệu từ các cảm biến để đưa ra các tín hiệu gửi tới màn hình.

Diagram

Description automatically generated

### State Diagram

Biểu đồ trạng thái (state diagram) để hiện luồng hoạt động của một thành phần hệ thống. Dưới đây là biểu đồ trạng thái của phần mềm hệ thống.

Diagram

Description automatically generated

Hình . State diagram của hệ thống

### Sequence Diagram

Sơ đồ trình tự (Sequence diagram) thể hiện tương tác giữa một tập hợp các đối tượng của hệ thống. Sơ đồ trình tự bao gồm tất cả các thông báo cho một phần hoặc một trường hợp sử dụng. Ở đây, nhóm sử dụng sơ đồ trình tự để thể hiện sự tương tác giữa phần cứng và phần mềm, và các tương tác cho trường hợp thông báo của hệ thống.

#### Sequence Diagram cho hệ thống xả nước

Diagram

Description automatically generated

Hình . Sequence diagram của hệ thống xả

Dựa vào sơ đồ trên, khi người dùng bấm nút xịt, phần mềm hệ thống sẽ nhận tín hiệu. Sau đó control thự hiện điều khiển hệ thống vòi xịt cho người dùng. Sau khi xong sẽ có hệ thống đèn thông báo đã thực hiện xong việc xả nước cho người dùng.

#### Sequence Diagram cho hệ thống sấy khô

Diagram

Description automatically generated

Hình . Sequence diagram của hệ thống sấy khô

Sau khi kết thúc quá trình xả nước , control thực hiện điều khiển hệ thống sấy khô bát đĩa. Sau khi sấy khô xong sẽ có hệ thống đèn thông báo đã thực hiện xong việc sấy khô cho người dùng.

#### Sequence Diagram cho hệ thống thải nước tự động

Diagram

Description automatically generated

Hình . Sequence diagram của hệ thống thải nước tự động

Khi quá trình rửa đã xong, cảm biến sẽ gửi tín hiệu tới phần mềm. Lúc này hệ thống bắt đầu thực hiện đếm thời gian tới khi cảm biến gửi tín để đưa ra chế độ xả nước phù hợp gửi tới control. Sau đó, control gửi tín hiệu xuống hệ thống xả nước của bồn cầu.

#### Sequence Diagram cho hệ thống làm ấm tự động

Diagram

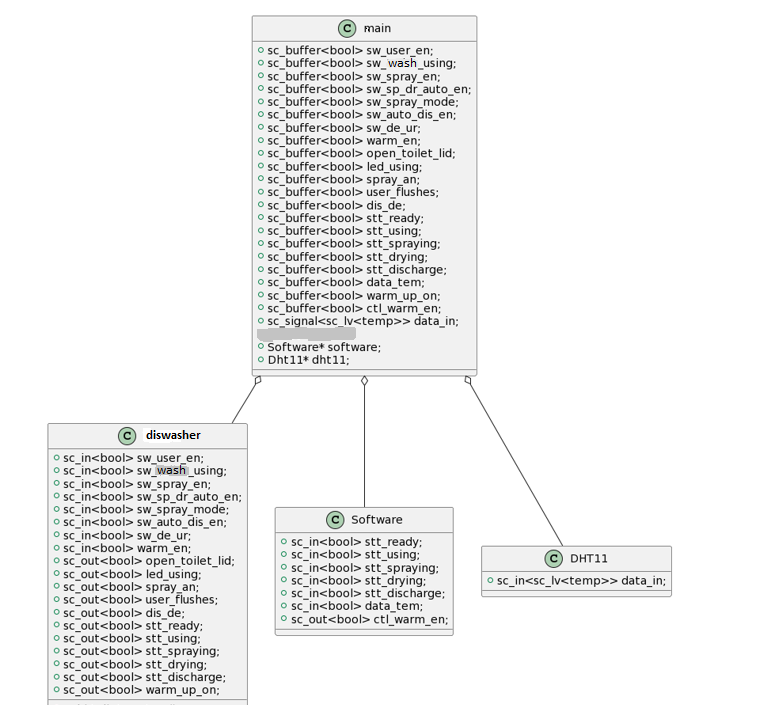
Description automatically generated

Hình . Sequence diagram của hệ thống làm ấm tự động

Cảm biến cảm nhận nhiệt độ từ môi trường rồi gửi tín hiệu tới phần mềm. Lúc này phần mềm sẽ so sánh nhiệt độ nhận được từ cảm biến với các ngưỡng đã thiết lập để đưa ra điều chỉnh về nhiệt độ phù hợp. Sau đó, tín hiệu này được gửi tới control để điều khiển hệ thống làm ấm.

## Mô hình hóa hệ thống sử dụng SystemC

Ngoài việc sử dụng UML, nhóm còn chọn systemC là ngôn ngữ thứ hai để thiết kế và mô hình hóa hệ thống bồn cầu tự động . Sơ đồ hệ thống của hệ thống rửa bát tự động được mô tả trên hình dưới đây.



Hình . Sơ đồ hệ thống rửa bát tự động sử dụng SystemC

Dựa vào yêu cầu kĩ thuật ở phần trước, nhóm sử dụng SystemC để thiết kế các đầu vào và đầu ra của hệ thống rửa bát tự động được thể hiện như trên hình gồm các class sau:

* Class main chứa các tín hiệu đầu vào đầu ra của hệ thống phục vụ cho việc mô phỏng.
* Class control chứa phần điều khiển hệ thống.
* Class Software chứa tín hiệu về trạng thái của các chân chức năng.
* Class DHT11 chứa tín về nhiệt độ, độ ẩm.

Bảng . Tín hiệu đầu vào/đầu ra của control

|  |  |  |
| --- | --- | --- |
| **Tên** | **Input/Output** | **Mô tả** |
| sw\_user\_en | input | Cảm biến khoảng cách vật |
| sw\_wash\_using | input | Cảm biến trọng lực |
| sw\_spray\_en | input | Nút bấm xịt nước |
| sw\_sp\_dr\_auto\_en | input | Nút bấm bật/tắt chế độ tự động xịt sấy |
| sw\_spray\_mode | input | Nút bấm chọn kiểu xịt |
| sw\_auto\_dis\_en | input | Nút bấm bật/tắt tự động xả |
| sw\_de\_ur | input | Cảm biến xác định đi nặng/đi nhẹ |
| warm\_en | input | Cảm biến nhiệt độ |
| open\_lid | output | Tín hiệu mở cửa |
| led\_using | output | Tín hiệu thông báo đang sử dụng |
| spray\_an | output | Tín hiệu chọn chế độ xịt |
| user\_flushes | output | Tín hiệu thông báo tự xả |
| stt\_ready | output | Tín hiệu trạng thái sẵn sàng |
| stt\_using | output | Tín hiệu trạng thái đang sử dụng |
| stt\_spraying | output | Tín hiệu trạng thái đang xịt nước |
| stt\_drying | output | Tín hiệu trạng thái đang sấy khô |
| stt\_discharge | output | Tín hiệu trạng thái đang xả nước |
| warm\_up\_on | output | Tín hiệu mở hệ thống làm ấm nước |

Bảng . Tín hiệu đầu vào/đầu ra của phần mềm hệ thống

|  |  |  |
| --- | --- | --- |
| **Tên** | **Input/Output** | **Mô tả** |
| stt\_ready | input | Tín hiệu trạng thái sẵn sàng |
| stt\_using | input | Tín hiệu trạng thái đang sử dụng |
| stt\_spraying | input | Tín hiệu trạng thái đang xịt nước |
| stt\_drying | input | Tín hiệu trạng thái đang sấy khô |
| stt\_discharge | input | Tín hiệu trạng thái đang xả nước |
| data\_tem | input | Dữ liệu nhiệt độ |
| ctl\_warm\_en | output | Đưa tín hiệu vào khối thanh ghi |

Bảng . Tín hiệu đầu vào/đầu ra của DHT11

|  |  |  |
| --- | --- | --- |
| **Tên** | **Input/Output** | **Mô tả** |
| data\_in | inout | Tín hiệu từ cảm biến nhiệt độ |

## Mô hình bằng FSM

### Tổng quan về FSM

Máy trạng thái hữu hạn, viết tắt là FSM, là một thành phần được sử dụng phổ biến trong thiết kế vi mạch số với ưu điểm là dễ kiểm soát quá trình hoạt động của thiết kế và dễ debug hoạt động của thiết kế.

|  |
| --- |
| [Diagram  Description automatically generated](https://1.bp.blogspot.com/-M8hoK35DiXI/XcgH3Cj04ZI/AAAAAAAALaY/vVVZwyQ6cWAOxufaS3DztYbX6VL16RmQwCKgBGAsYHg/s1600/image.png) |
| Mô hình cơ bản của FSM |

 FSM gồm có 3 thành phần cơ bản như sau:

1. Mạch tạo trạng thái kế tiếp (Next state logic) là mạch tổ hợp phụ thuộc vào ngõ vào FSM và giá trị trạng thái hiện tại lấy từ bộ nhớ trạng thái (state memory)
2. Bộ nhớ trạng thái (state memory) là phần tử lưu trạng thái hiện tại của FSM nó có thể là Flip-Flop, Latch, ... lấy ngõ vào từ mạch tạo trạng thái kế tiếp. Bộ nhớ trạng thái thường được sử dụng trong các thiết kế đồng bộ là FF hoạt động theo xung clock. Một tín hiệu reset có thể phải sử dụng để khởi động FSM đến một giá trị ban đầu. Reset không cần sử dụng đối với các FSM luôn hoạt động đúng dù giá trị ban đầu của FF là bao nhiêu.
3. Mạch tạo ngõ ra (output logic) là mạch tổ hợp tạo giá trị ngõ ra tương ứng với trạng thái hiện tại của FSM. Mạch này lấy ngõ vào là giá trị trạng thái hiện tại và có thể tổ hợp thêm ngõ vào của FSM

**FSM được chia làm 2 loại:**

* FSM Moore là loại có mạch tạo ngõ ra không phụ thuộc trực tiếp vào ngõ vào FSM
* FSM Mealy là loại có mạch tạo ngõ ra phụ thuộc trực tiếp vào ngõ vào FSM

|  |
| --- |
|  |
| FSM Moore |

|  |
| --- |
|  |
| FSM Mealy |

* **So sánh FSM Moore và FSM Mealy**

FSM Moore có tính an toàn cao hơn FSM Mealy vì ngõ ra FSM được tổ hợp từ giá trị FF nên nếu được kết nối đến các khối khác thì tín hiệu này có thể được sử dụng làm đường hồi tiếp trở lại tổ hợp với các mạch tạo ngõ vào cho FSM. Còn đối với FSM Mealy thì việc nay sẽ tạo ra mạch bất đồng bộ.

|  |
| --- |
|  |
| *Sử dụng FSM Moore để tránh tạo mạch bất đồng bộ không mong muốn khi hồi tiếp ngõ ra* |

Độ trễ ngõ ra FSM Moore được xác định rõ ràng và giới hạn trong FSM tính từ FF đi qua mạch tổ hợp tạo ngõ ra. Trong khi ngõ ra FSM Mealy phụ thuộc cả vào ngõ vào nên độ trễ của các tầng trước nối đến ngõ vào FSM cũng được tính vào độ trễ ngõ ra FSM và có thể là đường critical path (đường có độ trễ lớn nhất).

|  |
| --- |
|  |
| *Độ trễ ngõ ra của FSM Mealy phụ thuộc vào độ trễ ngõ vào nhưng FSM Moore thì không* |

Mealy có ưu điểm là số trạng thái ít hơn Moore với cùng một chức năng và ngõ ra đáp ứng ngay theo sự thay đổi của ngõ vào mà không cần chờ đến cạnh lên xung clock. Nếu dùng Mealy thì cần đặc biệt chú ý đến vấn đề hồi tiếp và độ trễ cộng dồn như đã trình bày ở trên.

### Tổng quan về HDL

Với sự phát triển nhanh chóng của khoa học và công nghệ, việc thiết kế các hệ thống điện tử cũng tạo ra một sự thay đổi mang tính cách mạng, một loại công cụ phát triển mới liên quan đến các hệ thống điện tử đang lan rộng nhanh chóng. Ngôn ngữ mô tả phần cứng (HDL) là một phương pháp để mô tả mạch kỹ thuật số. Với sự phát triển nhanh chóng của tự động hóa thiết kế hệ thống điện tử (EDA) và khả năng lập trình quy mô lớn của thiết bị logic, HDL có mô tả và mô phỏng phân cấp của bất kỳ đặc điểm thành phần điện tử nào, để các nhà thiết kế và phát triển mạch có thể tự do mô tả tính năng của mạch.

Ngôn ngữ Verilog là một loại mức độ trừu tượng của ngôn ngữ mô tả phần cứng. Ngôn ngữ này hỗ trợ khái niệm thiết kế trừu tượng ban đầu và có thể hiện thực hóa thiết kế trừu tượng sau này. Nó bao gồm cấu trúc phân cấp, cho phép các nhà thiết kế mô tả mức độ phức tạp của điều khiển. Verilog HDL là một ngôn ngữ không chỉ dễ sử dụng mà còn có tính năng mạnh mẽ, đặc biệt là chuẩn hóa công nghiệp Verilog HDL, phù hợp với xu thế phát triển của công nghệ vi điện tử.

### Thiết kế máy rửa bát

* Nguyên lý hoạt động của máy rửa bát:

Các máy phân loại, rửa bát có các chức năng hoạt động cơ bản: (1) phân loại bát, đĩa…, (2) đổ nước, (3) rửa, (4) xả nước ra và (4) sấy khô.

Máy phân loại có băng tải sẽ đưa sản phẩm qua khu cảm biến, mỗi sản phẩm sẽ có xung tương ứng báo cho khối xử lý, từ đó con lăn tương ứng sẽ được kích hoạt để đưa sản phẩm vào đúng khu vực của nó;

Máy rửa có các trạng thái liên tiếp sau: không hoạt động, đổ đầy nước, thêm chất rửa, rửa, xả nước, đổ đầy nước, rửa sạch, xả nước, sấy khô.

### Chi tiết về bộ điều khiển máy rửa bát

Bộ điều khiển bao gồm hai khối: khối máy hữu hạn (FSM) và khối hẹn giờ. Khối FSM nhận một số tín hiệu từ người dùng, từ bộ hẹn giờ và từ các bộ phận phần cứng khác như cảm biến. Đầu ra của khối FSM điều khiển khối hẹn giờ và các thành phần phần cứng khác của máy. Khối hẹn giờ tạo ra các khoảng thời gian chính xác cần thiết cho mỗi chu kỳ sau khi nó được đặt lại. Khối hẹn giờ bao gồm bộ đếm và logic tổ hợp để đưa ra tín hiệu thời gian chính xác sau khi đạt được các giá trị đếm nhất định. Tất nhiên, các giá trị của bộ đếm thời gian sẽ được xác định bởi tần số đồng hồ đang được sử dụng trong hệ thống.

### Quy trình thiết kế

1.Thiết kế biểu đồ trạng thái cho máy

2. Liệt kê các trạng thái của sơ đồ chuyển đổi trạng thái

3. Thiết kế bộ đếm thời gian.

Trong bài tập này, chúng em thiết kế, mô hình hóa và mô phỏng bộ điều khiển máy phân loại, rửa bát.

• Sử dụng kiểu mã hóa FSM Verilog cho khối FSM.

• Viết một trường hợp thử nghiệm để xác minh hoạt động của máy.

### Thiết kế cho phần phân loại chén đĩa

Căn cứ trên hoạt động mà yêu cầu đề ra, sơ đồ khối của máy như sau:

1. TIME\_COUNTER: dùng để đếm xung clock xác định thời gian duy trì trạng thái của các motor. Đây là thời gian có thể cấu hình được trước khi biên dịch như yêu cầu đặt ra. Ba tín hiệu bowl\_end, plate\_end và chopstick\_end báo thời điểm kết thúc của trạng thái BOWL, PLATE, CHOPSTICK. Ba tín hiệu fsm\_bowl, fsm\_plate và fsm\_chopstick báo trạng thái BOWL, PLATE, CHOPSTICK để bộ đếm hoạt động theo giá trị cấu hình phù hợp.
2. FSM: là máy trạng thái sẽ tạo ngõ ra motor\_1, motor\_2, motor\_3, và fsm\_bowl, fsm\_plate và fsm\_chopstick.

Diagram

Description automatically generated

*Sơ đồ khối của bộ điều khiển phân loại bát, đĩa, đũa*

**Phân tích khối FSM**

Việc chuyển trạng thái chỉ xảy ra khi tín hiệu báo độ trễ tương ứng của trạng thái đó tích cực. Ví dụ, khi motor-1 đang bật thì bowl-end phải tích cực thì mới chuyển sang trạng thái tiếp theo.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Output State | motor-1 | motor-2 | motor-3 | fsm-bowl | fsm-plate | fsm-chopstick |
| IDLE | 0 | 0 | 0 | 0 | 0 | 0 |
| BOWL | 1 | 0 | 0 | 1 | 0 | 0 |
| PLATE | 0 | 1 | 0 | 0 | 1 | 0 |
| CHOPSTICK | 0 | 0 | 1 | 0 | 0 | 1 |

*Bảng giá trị ngõ ra FSM của máy phân loại.*

**Diagram

Description automatically generated**

*FSM điều khiển phân loại bát, đĩa, đũa.*

**Phân tích khối TIME/COUNTER**

Chú ý, bowl\_time, plate\_time và chopstick\_time là các hằng số được cấu hình trước khi biên dịch RTL code. Với tần số 1 Hz, nếu muốn thời gian chạy của motor\_1 là 2s, motor\_2 là 2s, và motor\_3 là 2s thì cần gán giá trị như sau:

* bowl\_time = 1
* plate\_time = 1
* chopstick\_time = 1

Diagram

Description automatically generated

*Sơ đồ nguyên lý khối TIME/COUNTER của máy phân loại.*

### Thiết kế cho phần rửa

Căn cứ trên hoạt động mà yêu cầu đề ra, sơ đồ khối của máy như sau:

TIME/COUNTER: dùng để đếm xung clock xác định thời gian duy trì trạng thái của các motor. Đây là thời gian có thể cấu hình được trước khi biên dịch như yêu cầu đặt ra. Hai tín hiệu wash\_timeout và drying\_timeout báo thời điểm kết thúc của trạng thái WASH và DRYING. Hai tín hiệu fsm\_wash và fsm\_drying báo trạng thái WASH và DRYING để bộ đếm hoạt động theo giá trị cấu hình phù hợp.

FSM: là máy trạng thái sẽ tạo ngõ ra door\_lock, motor\_on, fill\_valve\_on, drain\_valve\_on, done, và fsm\_wash, fsm\_drying.

Diagram

Description automatically generated

*Sơ đồ khối của máy rửa*.

**Phân tích khối FSM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| State Output | check\_door | fill\_water | add\_soap | wash | drain\_water | drying |
| door\_look | 0 | 1 | 1 | 1 | 1 | 1 |
| motor\_on | 0 | 0 | 0 | 1 | 0 | 0 |
| fill\_valve\_on | 0 | 1 | 0 | 0 | 0 | 0 |
| drain\_valve\_on | 0 | 0 | 0 | 0 | 1 | 0 |
| done | 1 | 0 | 0 | 0 | 0 | 0 |
| fsm\_wash | 0 | 0 | 0 | 1 | 0 | 0 |
| fsm\_drying | 0 | 0 | 0 | 0 | 0 | 1 |

*Bảng giá trị ngõ ra FSM của máy rửa.*

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

*FSM điều khiển phân loại bát, đĩa, đũa.*

**Diagram

Description automatically generated**

*Các trạng thái liên tiếp của máy rửa*.

**Phân tích khối TIME/COUNTER**

Chú ý, WASH\_TIME và DRYING\_TIME là các hằng số được cấu hình trước khi biên dịch RTL code. Với tần số 1 Hz, nếu muốn thời gian rửa là 300s và thời gian sấy khô là 100s thì cần gán giá trị như sau:

* WASH\_TIME = 299
* DRYING\_TIME = 99

Diagram

Description automatically generated

*Sơ đồ nguyên lý khối TIME/COUNTER của máy rửa*.

### Mô phỏng

* Mô phỏng phần phân loại

Ảnh có chứa văn bản, thiết bị điện tử, ảnh chụp màn hình, máy tính

Mô tả được tạo tự động

* Mô phỏng phần rửa bát

Graphical user interface

Description automatically generated

*Mô phỏng bộ điều khiển máy rửa.*

Khi có tín hiệu reset, tất cả các tín hiệu được đặt thành 0; sau đó nếu nhấn nút start, máy sẽ tự động thực hiện các công đoạn phân loại và rửa theo quy trình định sẵn. Kết quả mô phỏng như trong hình.

## Kết luận

Trong chương này, nhóm đã trình bày việc mô hình hóa hệ thống bồn cầu tự động sử dụng hai công cụ là UML, SystemC và FSM. Từ đó, nhóm có thể hình dung tổng quan về hệ thống và các phần cần thiết để đáp ứng chỉ tiêu kĩ thuật đã nêu ra ở CHƯƠNG 1. Công việc triển khai hệ thống được trình bày ở chương tiếp theo.

Chúng em sử dụng ngôn ngữ Verilog HDL để thiết kế hệ thống điều khiển máy phân loại, rửa bát. Thông qua việc hiểu mạch phần cứng tương ứng và các công cụ của ngôn ngữ Verilog HDL để tạo ra nhiều hơn phương pháp thiết kế logic truyền thống có thể thích ứng với nhu cầu phát triển của xã hội. Chúng em sử dụng ngôn ngữ mô tả phần cứng dưới dạng thiết kế hệ thống kỹ thuật số không chỉ linh hoạt và thuận tiện mà còn giảm chi phí phát triển và chu kỳ phát triển. Phương pháp thiết kế này đóng một vai trò ngày càng quan trọng trong thiết kế hệ thống kỹ thuật số trong tương lai.

# THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN)

Chương này mô tả thiết kế chi tiết từng khối và toàn hệ thống máy rửa bát tự động.

## Thiết kế phần cứng

### Mô tả chức năng

Như chúng ta đã biết, khi chúng ta bắt đầu một máy rửa chén, nhiều điều xảy ra; Ví dụ, lượng nước nạp, vận hành động cơ xoáy nước để làm sạch bát đĩa, thoát nước, v.v. Máy rửa chén cũng cung cấp nhiều chu trình rửa khác nhau để xử lý các loại dụng cụ khác nhau và lượng làm sạch mà chúng cần. Ngoài ra, nếu bạn mở cửa máy rửa chén ở giữa hoạt động của nó, rất có thể nó sẽ cảm thấy rằng cánh cửa được mở và sẽ ngừng hoạt động. Và các máy rửa chén cao cấp mới nhất thậm chí có thể có màn hình LCD hiển thị nhiều trạng thái và thông tin điều khiển khác nhau trên bảng điều khiển phía trước

**Mô tả hoạt động của bồn cầu khi có người sử dụng**:

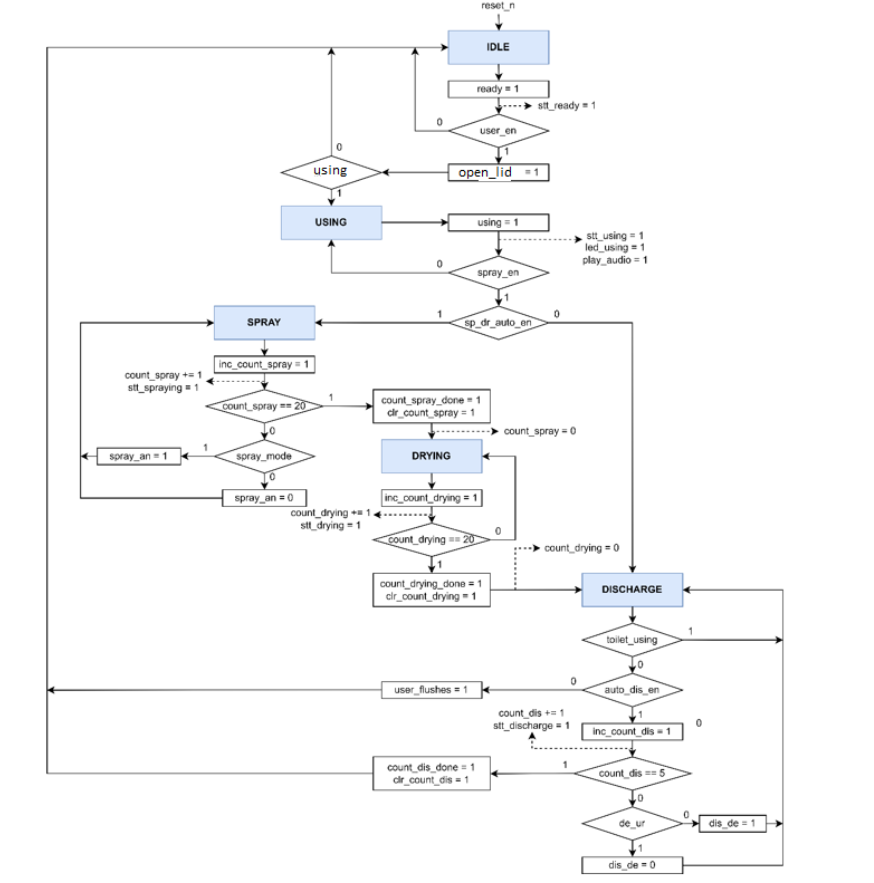
Nguyên lý hoạt động của máy rửa bát là sử dụng lực phun xoáy mạnh của nước lạnh hoặc nước nóng. Hỗn hợp nước sạch và chất tẩy rửa tự động phun nước thông qua hệ thống phun áp lực cao để làm sạch vết bẩn dính trên vật dụng.  
Trong khi đó các giàn rửa bên trong xoay để thay đổi vị trí của đồ cùng nhằm tăng khả năng tiếp xúc. Điều này khác hoàn toàn so với việc rửa bằng tay thông thường khi phải dùng sức để rửa sạch hoàn toàn dầu mỡ bám trên chén đĩa.

Tiếp đó, máy rửa bát sẽ hấp khô dụng cụ bếp của bạn bằng cách tạo luồng khí nóng để làm khô bát đĩa hiệu quả. Thời gian một quy trình rửa khoảng 25 - 35 phút. Lượng điện năng tiêu thụ khoảng 1 - 3 kWh và lượng nước tiêu thụ khoảng 20 - 30 lít để hoàn thành chu trình. Qua nguyên lý hoạt động máy trên có thể thấy, công việc rửa chén không tốn điện và nước như ta vẫn nghĩ.

### Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath)

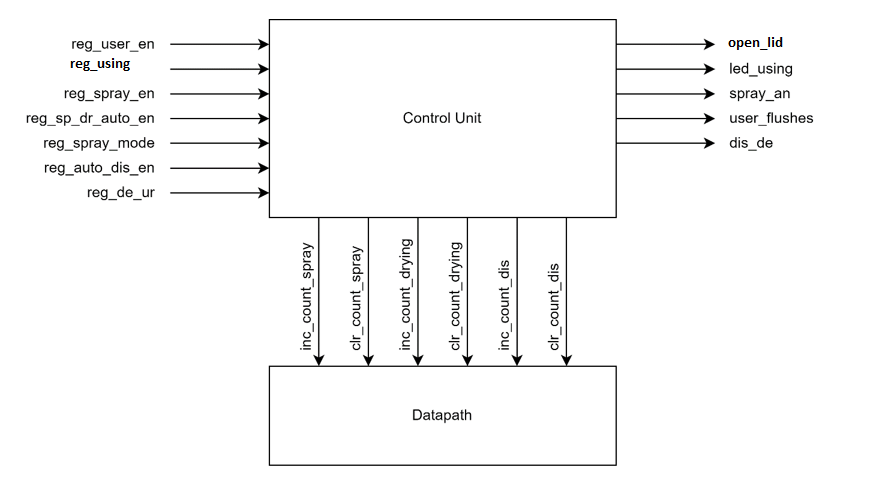
Từ mô tả chức năng và hoạt động của hệ thống máy rửa tự động, xác định được năm trạng thái:

* IDLE: trạng thái IDLE, hệ thống sẽ trong trạng thái chờ người sử dụng.
* USING: trạng thái đang sử dụng.
* SPRAY: trạng thái xịt nước.
* DRYING: trạng thái sấy khô.
* DISCHARGE: trạng thái xả nước.



Hình . Sơ đồ ASMD

Từ sơ đồ ASMD ta xác định được sơ đồ khối giao tiếp giữa control unit và datapath của hệ thống:



Hình . Sơ đồ khối Control Unit và Datapath

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên tín hiệu** | **Độ rộng (bit)** | **Chiều** | **Mô tả** |
| 1 | reg\_user\_en | 1 | input | Cảm biến khoảng cách vật |
| 2 | reg\_using | 1 | input | Cảm biến trọng lực |
| 3 | reg\_spray\_en | 1 | input | Nút bấm xịt nước |
| 4 | reg\_sp\_dr\_auto\_en | 1 | input | Nút bấm bật/tắt chế độ tự động xịt sấy |
| 5 | reg\_spray\_mode | 1 | input | Nút bấm chọn kiểu xịt |
| 6 | reg\_auto\_dis\_en | 1 | input | Nút bấm bật/tắt tự động xả |
| 8 | open\_lid | 1 | output | Tín hiệu mở nắp máy |
| 9 | led\_using | 1 | output | Tín hiệu thông báo đang sử dụng |
| 10 | spray\_an | 1 | output | Tín hiệu chọn chế độ xịt |
| 11 | user\_flushes | 1 | output | Tín hiệu thông báo người dùng tự xả |
| 12 | dis\_de | 1 | output | Tín hiệu chọn chế độ xả nước |

### Thiết kế khối register\_block cho system\_controller

A picture containing graphical user interface

Description automatically generated

Hình . Các thanh ghi trong system\_controller

Thiết kế khối register block có giao tiếp APB được mô tả bằng ngôn ngữ verilog.

### Thiết kế RTL

Phần thiết kế được mô tả trong source code tại thư mục *hdl.*

Các thiết kế bao gồm:

* clock\_generator
* counter
* warm\_up
* controller
* core
* smart\_wash\_system\_top

## Thiết kế phần mềm

Trình bày tổng quát về phần mềm thiết kế, cũng như có 1 cái nhìn tổng thể về hệ thống, những vấn đề phát sinh và hướng giải quyết khi gặp vấn đề và đưa ra giải pháp thay thế với những bài toán đã đặt ra.

### Mục tiêu

Nhóm đề xuất việc áp dụng phần mềm vào việc hiển thị thông báo và xử lý dữ liệu nhận về từ các sensors.

### Lý do lựa chọn phần mềm

Lý do nhóm lựa chọn phần mềm để thực hiện mục đích này gồm những ý chính sau:

* Thứ nhất: Việc để phần mềm xử lý logic sẽ dễ thực hiện hơn phần cứng, mặc dù về chi phí thì cũng không quá tốn kém hơn việc sử dụng phần cứng.
* Thứ hai: Việc sử dụng phần mềm giúp chúng ta có thể bảo trì hệ thống dễ dàng hơn khi sử dụng nhiều cảm biến.
* Cuối cùng: Tính tùy biến cho phần mềm cao hơn, thay vì việc thiết kế FPGA thì đối với việc thay đổi thuật toán trở nên vô cùng khó khăn, gần như việc đổi thuật toán nếu hệ thống đã thương mại hóa thì gần nhưng chúng ta phải thay thế con chip FPGA, nhưng đối với phần mềm, khi đưa vào vận hành, nếu thấy thuật toán đang có vấn đề, chúng ta cần tùy biến thì chúng ta chỉ cần nạp lại code, là hệ thống có thể thay đổi các vận hành. Điều này làm nên sức mạnh của hệ thống phần mềm.

### Thiết kế

Diagram

Description automatically generated

Diagram

Description automatically generated

Hình . Sơ đồ phần mềm hệ thống

**Giải thích:**

* Phần mềm sẽ liên tục đọc các dữ liệu qua các thanh ghi từ khối điều khiển cảm biến để xác định trạng thái của môi trường.
* Dựa vào các dữ liệu đọc được đưa ra tín hiệu điều khiển cho khối điều khiển bồn cầu.
* Đọc các trạng thái từ các thanh ghi của khối điều khiển bồn cầu, hiển thị lên màn hình OLED.

### Hướng triển khai

Việc triển khai được sử dụng ngôn ngữ lập trình C, lập trình trên phần mềm Vivado

Sử dụng hệ điều hành FreeRTOS (Real-time operating system). Đồng thời việc triển khai được viết trên ngôn ngữ C.

Các hàm cơ bản sử dụng:

|  |
| --- |
| static INLINE u32 Xil\_In32(UINTPTR Addr); |

Hàm đọc 32 bit dữ liệu từ địa chỉ Addr

|  |
| --- |
| static INLINE void Xil\_Out32(UINTPTR Addr, u32 Value); |

Hàm ghi 32 bit Value vào địa chỉ Addr

### Sơ đồ thuật toán

Khi sử dụng hệ điều hành FreeRTOS, chúng ta sẽ chạy được đồng thời nhiều task và AXI sẽ chia ra các kênh đọc ghi riêng nên ở đây chúng ta sẽ chia làm 2 task:

* Đọc dữ liệu (xReadTask)
* Ghi dữ liệu (xWriteTask)

Diagram

Description automatically generated

Hình . Sơ đồ thuật toán

Hình 3.5 mô tả sơ đồ thuật toán của phần mềm.

Ở đây chúng ta sửa dụng các hàm đọc ghi cấu hình các thanh ghi thông qua giao tiếp AXI để điều khiển các module. Các thanh ghi có địa chỉ và công dụng đã được mô tả chi tiết ở chương 4.

Diagram

Description automatically generated

Hình . Mô hình giao tiếp giữa vi xử lý và các module

Hình 3.6 mô tả sơ bộ toàn hệ thống giao tiếp giữa phần vi xử lí và phần cứng tự thiết kế. Trong project này, chúng em chọn giao tiếp GPIO để truyền thông giữa phần mềm và phần cứng. Để đồng bộ giữa 2 phần, chúng em thiết kế thêm một khối trung gian là register block hoạt động cùng tần số với tần số của controller, để captrure tín hiệu điều khiển được gửi từ phần mềm, gửi đầu vào ổn định cho controller và lưu trạng thái hiện tại của controller để phần mềm xử lý hiển thị.

# TRIỂN KHAI TRÊN FPGA (IMPLEMENT FPGA)

Trong chương này, chúng em sẽ trình bày về các bước triển khai hệ thống lên kit FPGA.

## Tổng quan về triển khai hệ thống trên FPGA

Kit sử dụng: ZedBoard (Xilinx Zynq®-7000 All Programmable SoC).

Part: 7z020clg484-1

Diagram, schematic

Description automatically generated

Hình .. Sơ đồ khối của hệ thống

Hệ thống sử dụng hệ vi xử lý Cortex™-A9 lõi kép được tích hợp sẵn trong chip Xilinx Zynq®-7000, từ đó thông qua giao thức AMBA – AXI (Advanced Microcontroller Bus Architecture - Advanced eXtensible Interface) đọc hoặc ghi dữ liệu ở các thanh ghi đã cấu hình sẵn.

Hệ thống sử dụng màn hình Oled đã có sẵn trên kit để hiển thị các trạng thái, nhiệt độ và các cảnh báo vì vậy cần 1 khối oled\_controller để điều khiển màn hình sử dụng giao thức SPI và giao tiếp AXI với vi xử lý.

Một khối dht11\_controller để điều khiển cũng như nhận dữ liệu từ cảm biến DHT11 với chuẩn giao tiếp TTL, 1 wire.

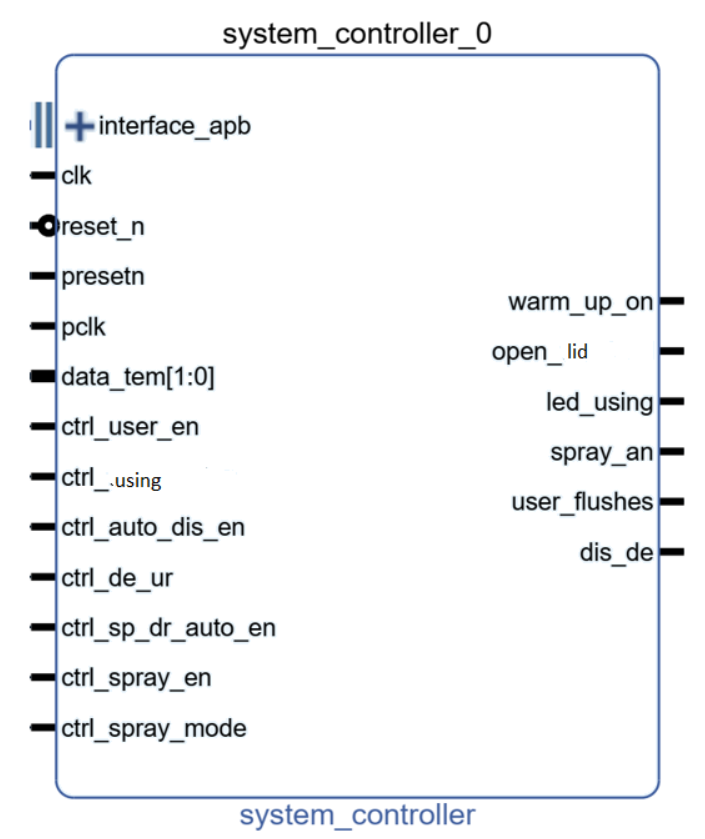
Khối system\_controller là khối chính của hệ thống, điều khiển hoạt động của máy (chương 3). Do thiết kế sử dụng giao tiếp AMBA – APB (Advanced Microcontroller Bus Architecture - Advanced Peripheral Bus) nên chúng ta cần AXI APB Bridge để giao tiếp với vi xử lý.

Các chân in, out sẽ được gán ràng buộc vào các LED, SWITCH, BUTTON và GPIO có sẵn trên kit để thuận tiện mô phỏng cũng như kiểm thử.

## Các IP dùng trong hệ thống

### Smart system controller (system\_controller)

Dựa vào phần thiết kế ở chương 3 bao gồm: khối Register block và khối System controller được triển khai bằng ngôn ngữ verilog. Sử dụng phần mềm Vivado để đóng gói thành IP system\_controller.



Hình . Sơ đồ khối của smart system controller

Dưới đây là bảng gán ràng buộc cho các tín hiệu của system\_controller lên các chân của FPGA

Bảng . Ràng buộc các tín hiệu của system\_controller

|  |  |  |
| --- | --- | --- |
| **Tín hiệu** | **Cổng** | **Mô tả** |
| warm\_up\_on | T22 | LED0 |
| open\_lid | T21 | LED1 |
| led\_using | U22 | LED2 |
| spray\_an | U21 | LED3 |
| dis\_de | V22 | LED4 |
| ctrl\_using | H22 | SW2 |
| ctrl\_spray\_mode | F21 | SW3 |
| ctrl\_spray\_en | H19 | SW4 |
| ctrl\_sp\_dr\_auto\_en | H18 | SW5 |
| ctrl\_de\_ur | H17 | SW6 |
| ctrl\_auto\_dis\_en | M15 | SW7 |
| user\_flushes | Y11 | JA1 |

Bảng 4.1 Ràng buộc các tín hiệu của system\_controllerchỉ ra các tín hiệu được constraint với các cổng trên kit zedboard. Các mô tả sẽ được chú thích trên kit.

### Oled controller (oled\_controller)

IP oled\_controller sử dụng giao thức SPI để giao tiếp với màn hình oled đã được gắn sẽ trên kit. Người dùng sẽ cấu hình các giá trị trong thanh ghi () để điều khiễn cũng như hiện thị dữ liệu lên màn hình.

A picture containing text

Description automatically generated

Hình . Sơ đồ khối của oled\_controller

Địa chỉ cơ sở (Base address): 0x43C00000

Luồng hoạt động:

* Ghi dữ liệu vào thanh ghi data (0x08) là 1 byte tương ứng với ký tự muốn in lên màn hình
* Kích hoạt IP sử dụng thanh ghi ctrl (0x00)
* Kiểm tra thanh ghi status (0x04) nếu mà bằng 0 thì sẽ chuyển sang trạng thái “polling mode”, sau khi bằng 1 tiến hành cấu hình lại thanh ghi status về 0.

Màn hình hiển thị tối đa 64 ký tự.

Bảng . Tổ chức thanh ghi trong oled\_controller

|  |  |  |  |
| --- | --- | --- | --- |
| **Thanh ghi** | **Mô tả** | **Địa chỉ** | **Mô tả của thanh ghi** |
| data | Thanh ghi dữ liệu | 0x08 | Ký hiệu cần ghi lên màn hình (1 char – 8bit) |
| ctrl | Thanh ghi điều khiển | 0x00 | Kích hoạt bộ điều khiển  1: Bật  0: Tắt |
| stt | Thanh ghi trạng thái | 0x04 | Kiểm tra trạng thái để quyết định ghi dữ liệu vào thanh ghi data |

Bảng . Ràng buộc các tín hiệu của oled\_controller

|  |  |  |
| --- | --- | --- |
| **Tín hiệu** | **Cổng** | **Mô tả** |
| oled\_dc\_n | U10 | OLED\_DC |
| oled\_reset\_n | U9 | OLED\_RES |
| oled\_spi\_clk | AB12 | OLED\_SCLK |
| oled\_spi\_data | AA12 | OLED\_SDIN |
| oled\_vbat | U11 | OLED\_VBAT |
| oled\_vdd | U12 | OLED\_VDD (GND) |

### DHT11 controller (dht11\_controller)

Cảm biến độ ẩm và nhiệt độ DHT11 Temperature Humidity Sensor là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp TTL - 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào. So với cảm biến đời mới hơn là DHT22 thì DHT11 cho khoảng đo và độ chính xác kém hơn rất nhiều.

Graphical user interface, text, application, chat or text message

Description automatically generated with medium confidence

Hình . Sơ đồ khối của DHT11 controller

Địa chỉ cơ sở (base address): 0x43C20000

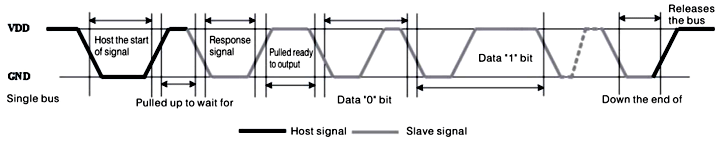
Bộ dht11\_controller điều khiển và xử lý dữ liệu nhận được từ cảm biến DHT11 sau đó dữ liệu đã xử lý được ghi vào các thanh ghi. Từ đó, vi xử lý có thể đọc nhiệt độ, độ ẩm từ các thanh ghi qua giao thức AXI.

Bảng . Tổ chức thanh ghi trong dht11\_controller

|  |  |  |
| --- | --- | --- |
| **Thanh ghi** | **Mô tả** | **Địa chỉ** |
| reg\_ctrl | Thanh ghi điều khiển | 0x00 |
| reg\_debug | Thanh ghi debug | 0x04 |
| reg\_temperature | Thanh ghi dữ liệu nhiệt độ | 0x08 |
| reg\_humidity | Thanh ghi dữ liệu độ ẩm | 0x0C |
| reg\_crc | Thanh ghi Cyclic Redundancy Check | 0x10 |
| reg\_ack | Thanh ghi phản hồi | 0x14 |

Bảng . Ràng buộc các tín hiệu của dht11\_controller

|  |  |  |
| --- | --- | --- |
| **Tín hiệu** | **Cổng** | **Mô tả** |
| data | AA11 | JA2 |



Hình . Sơ đồ thời gian quá trình gửi tín hiệu điều khiển từ master và phản hồi tín hiệu từ slave của cảm biến DHT11

## Tổng hợp và triển khai trên phần mềm Vivado

Trong phần nhóm sử dụng phần mềm Vivado v2018.1 (64-bit).

### Tạo Block design

Từ các IP được thiết kế và các IP có sẵn của Xilinx, block design được thiết kế như hình dưới.

Diagram, schematic

Description automatically generated

Hình . Block design của hệ thống

### Tạo constraint file

Tạo file ràng buộc nối các input và output của block design với các chân của kit FPGA (chi tiết xem tại ZedBoard constraint file)

Dưới đây là ví dụ contraint các chân của oled controller với port trên kit.

|  |
| --- |
| #OLED  set\_property PACKAGE\_PIN U10 [get\_ports oled\_dc\_n];  set\_property PACKAGE\_PIN U9 [get\_ports oled\_reset\_n];  set\_property PACKAGE\_PIN AB12 [get\_ports oled\_spi\_clk];  set\_property PACKAGE\_PIN AA12 [get\_ports oled\_spi\_data];  set\_property PACKAGE\_PIN U11 [get\_ports oled\_vbat];  set\_property PACKAGE\_PIN U12 [get\_ports oled\_vdd];  set\_property IOSTANDARD LVCMOS33 [get\_ports oled\_dc\_n];  set\_property IOSTANDARD LVCMOS33 [get\_ports oled\_reset\_n];  set\_property IOSTANDARD LVCMOS33 [get\_ports oled\_spi\_clk];  set\_property IOSTANDARD LVCMOS33 [get\_ports oled\_spi\_data];  set\_property IOSTANDARD LVCMOS33 [get\_ports oled\_vbat];  set\_property IOSTANDARD LVCMOS33 [get\_ports oled\_vdd]; |

### Tổng hợp (systhesis)

Kết quả của quá trình tổng hợp được thể hiện tỏng bảng sau:

Bảng . Số lượng LUT và Register của thiết kế

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Loại** | | | **Sử dụng** | **Tỷ lệ** |
| Slice LUTs |  | | 1708 | 3.21 |
| LUT as Logic | | 1642 | 3.09 |
| LUT as Memory | LUT as Distributed RAM | 66 | 0.38 |
| LUT as Shift Register | 0 |  |
| Slice Registers |  | | 2297 | 2.16 |
| Register as Flip Flop | | 2265 | 2.13 |
| Register as Latch | | 32 | 0.03 |
| F7 Muxes |  | | 108 | 0.41 |
| F8 Muxes |  | | 38 | 0.29 |

### Triển khai (implement)

Hình 4.8 cho ta thấy, thiết kế thỏa mãn setup time và hold time để đảm bảo slack dương. Kết quả timing tốt.

Graphical user interface, text, application

Description automatically generated

Hình . Báo cáo về timing

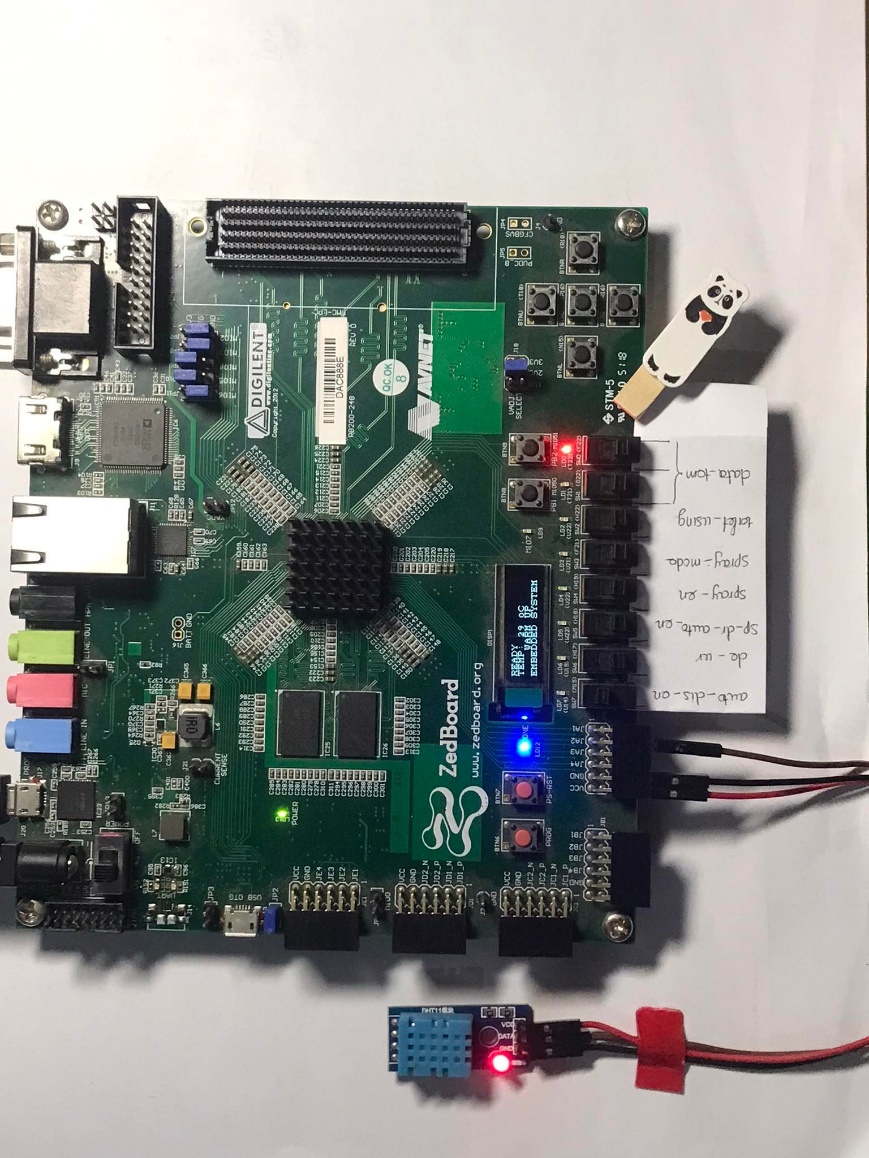
Hình 4.9 cho ta thấy kết quả về mặt công suất của thiết kế. Ở đây tổng công suất không tính công suất dự trữ (margin) là

Graphical user interface, application

Description automatically generated

Hình . Báo cáo về công suất của thiết kế

### Kết quả



Hình . Kết quả trên kit FPGA

Sử dụng các botton, switch, led, oled đã được contrain sẵn. Tiến hành thử những kịch bản khác nhau.

Kết quả: hệ thống đã chạy đúng với function đề ra ở khâu thiết kế. Về mặt timing thỏa mãn (độ trễ) và năng lượng tiêu tốn thấp trong phạm vi phù hợp.

# KIỂM THỬ (VERIFICATION)

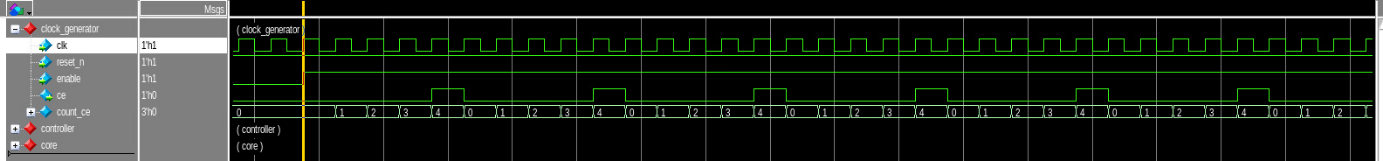
Chương này trình bày các kết quả mô phỏng cho từng khối và toàn bộ thiết kế được triển khai bằng ngôn ngữ Verilog và ngôn ngữ C trên phần mềm QuestaSim và Vivado.

## Kế hoạch kiểm thử

### Kiểm thử thiết kế phần cứng máy rửa bát sử dụng QuestaSim

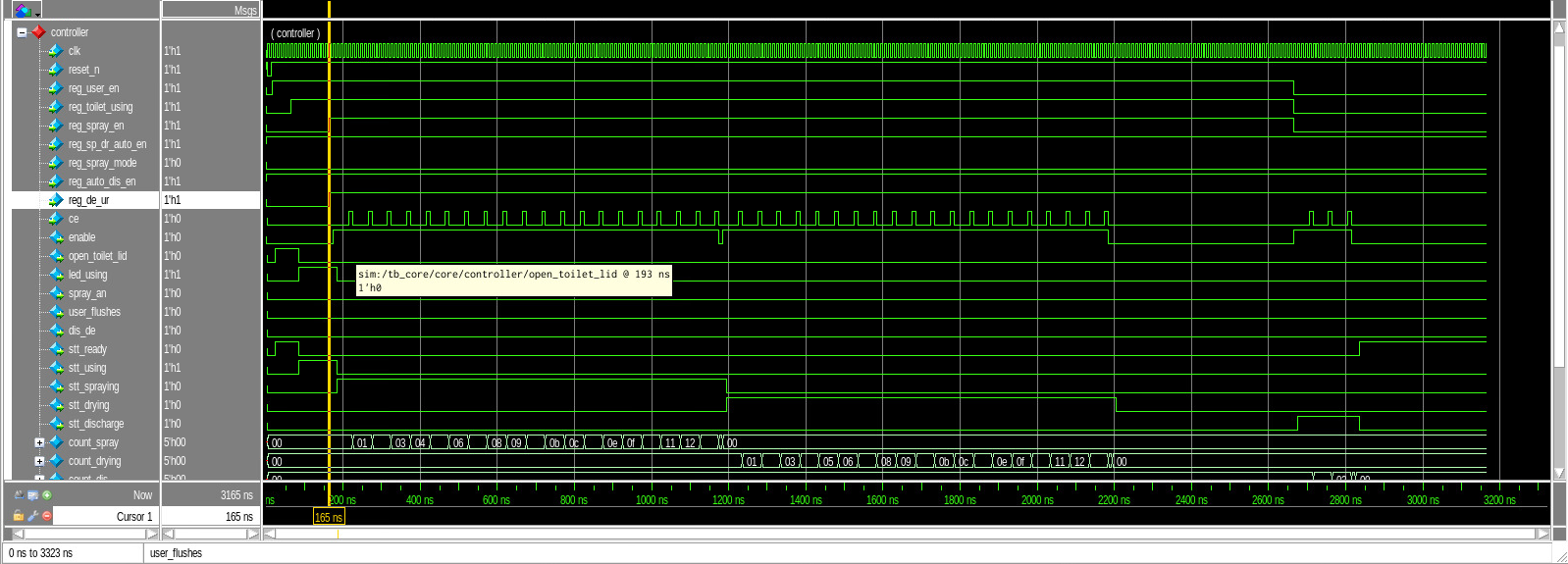
Xây dựng testbench dựa trên kịch bản test và kiểm tra kết quả đầu ra của khối controller sử dụng Verilog và QuestaSim

#### Khối clock\_generator



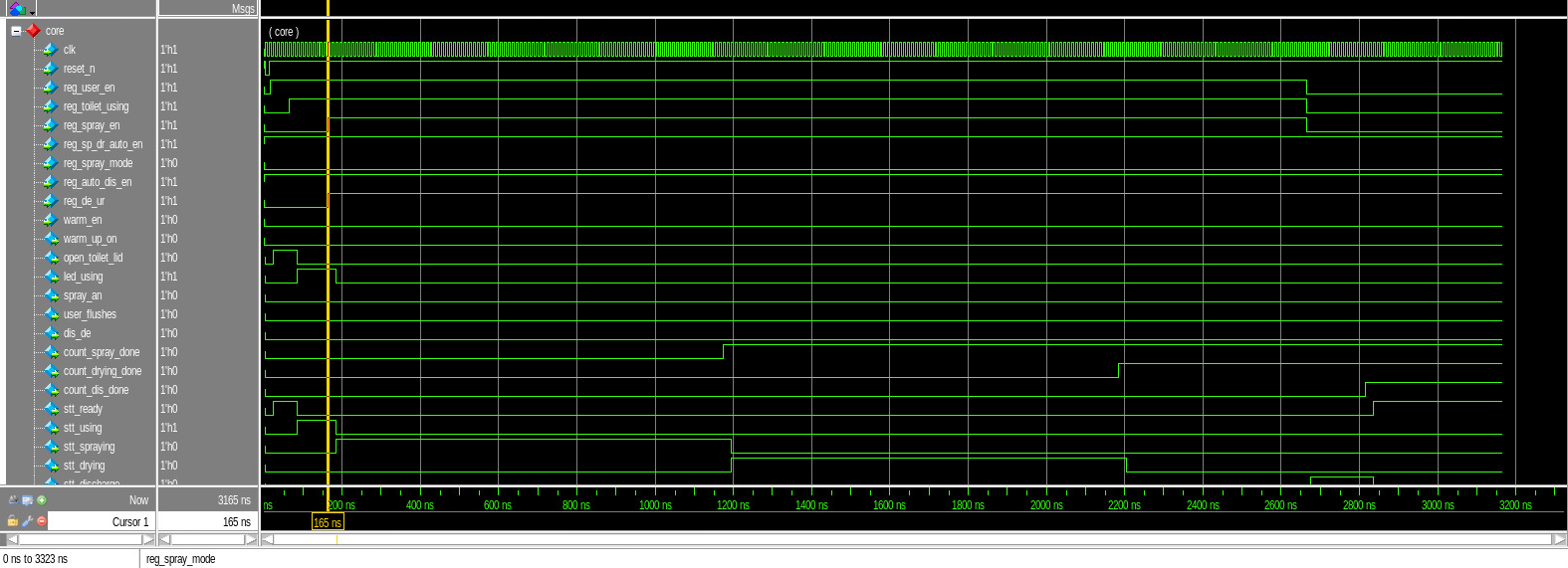
Hình . Kết quả mô phỏng khối clock\_generator

#### Khối controller



Hình . Kết quả mô phỏng khối controller

#### Khối core



Hình . Kết quả mô phỏng khối core

## Mô phỏng trên proteus

Chart

Description automatically generated

Ở đây chúng em sử dụng vi điều khiển 8051 để mô phỏng và dùng phần mềm kelic để biên dịch code

# KẾT LUẬN

Sau một thời gian nghiên cứu về hệ thống nhúng, đặc biệt là hệ thống nhúng máy rửa bát tự động, nhóm em đã nắm được các quy trình thiết kế một sản phẩm nhúng cũng như cấu tạo của máy và hoạt động của nó. Đồng thời nhóm đã thiết kế hệ thống điều khiển máy rửa tự động trên Xilinx ZedBoard. Hệ thống gồm các sensor, button sẽ nhận tín hiệu và đưa vào bộ điều khiển được viết bằng ngôn ngữ C, sau đó các tín hiệu sẽ được gửi tới bộ điều khiển được mô tả bằng ngôn ngữ Verilog để điều khiển hoạt động của buồn cầu.

Trong quá trình thực hiện do thời gian và kiến thức còn nhiều sai xót, mong thầy và các bạn đóng góp ý kiến để nhóm em hoàn thiện hơn.

Nhóm em xin chân thành cám ơn TS. Phạm Văn Tiến đã giúp đỡ nhóm chúng em trong quá trình thực hiện bài tập lớn này.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | https://www.answers.com/electrical-engineering/How\_embedded\_systems\_is\_used\_in\_dishwasher |
| [2] | https://www.sensiblemicro.com/blog/embedded-systems |
| [3] | https://texgio.vn/tin-tuc/may-rua-bat-la-gi.html |