

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO
ĐỒ ÁN THIẾT KẾ LUẬN LÍ (CO3091)

Đề tài 8:

“Máy tính đơn giản”

Nhóm 05 - L03

GVHD: ThS. Huỳnh Phúc Nghị
SV thực hiện: Huỳnh Bùi Ngọc Khoa - 2211589 (*Leader*)
Lưu Văn Huy - 2211199
Vũ Trọng Khánh - 2211541
Nghiêm Lương Sơn - 2014373
Nguyễn Khắc Vinh - 2015063

TP. HỒ CHÍ MINH, 12/2024

Mục lục

1	Mở đầu	6
1.1	Giới thiệu đề tài	7
1.2	Thiết bị sử dụng	8
1.3	Công cụ sử dụng	13
1.4	Chức năng sản phẩm	15
2	Lý thuyết cơ bản	18
2.1	Đặc tả hệ thống	19
2.1.1	Nhập dữ liệu từ người dùng:	19
2.1.2	Xử lý dữ liệu bên trong hệ thống	20
2.1.3	Hiển thị kết quả	20
2.2	Những tính năng được ứng dụng trong thực tế	20
3	Thiết kế	23
3.1	Sơ đồ khối	24
3.1.1	Bộ xử lý trung tâm (ALU - Arithmetic Logic Unit):	25
3.1.2	Bộ điều khiển (Control Unit)	25
3.1.3	Hệ thống nhập liệu:	26
3.1.4	Hệ thống hiển thị:	26
3.1.5	Bộ nhớ tạm thời (Register):	26
4	Hiện thực	28
4.1	Sơ đồ khối	29



5 Kết luận	31
6 DANH MỤC TÀI LIỆU THAM KHẢO	33

Danh sách bảng

1	Member list & workload	5
---	----------------------------------	---

Danh sách hình vẽ

1.1	Hình ảnh mạch Arty Z7-20	8
1.2	Cấu trúc sơ đồ GPIO mạch ARTY Z7-20	12
3.1	Sơ đồ khối hệ thống	24
4.1	Sơ đồ khối hiện thực	29

No.	Fullname	Student ID	Problems	% done
1	Huỳnh Bùi Ngọc Khoa	2211589	- Hoàn thiện module BCD - Hiển thị số trên LED 7 đoạn - Tổng hợp code	100%
2	Lưu Văn Huy	2211199	- Sửa lỗi module - Debug trên FFGA - Demo mạch	100%
3	Vũ Trọng Khánh	2211541	- Hiện thực các phép toán - Test code trên mạch - Viết báo cáo	100%
4	Nguyễn Khắc Vinh	2015063	- Hàm cộng, trừ, nhân, chia và full adder	100%
5	Nghiêm Lương Sơn	2014373	- Hàm chia lấy dư, logic và half adder	100%

Bảng 1: Member list & workload

1

Mở đầu

Trong báo cáo này, chúng tôi sẽ trình bày các nội dung bao gồm giới thiệu đề tài, công cụ, thiết bị sử dụng và các chức năng của sản phẩm.

Nội dung

1.1	Giới thiệu đề tài	7
1.2	Thiết bị sử dụng	8
1.3	Công cụ sử dụng	13
1.4	Chức năng sản phẩm	15

1.1 Giới thiệu đề tài

Hiện nay, máy tính cầm tay đang được sử dụng rất phổ biến, có nhiều loại máy tính từ tính toán đơn giản đến các máy tính khoa học. Với sự phát triển của công nghệ, máy tính cầm tay không chỉ là công cụ để thực hiện các phép tính cơ bản như cộng, trừ, nhân, chia mà còn được thiết kế với nhiều tính năng vượt trội cho các mục đích chuyên biệt trong từng lĩnh vực khác nhau.

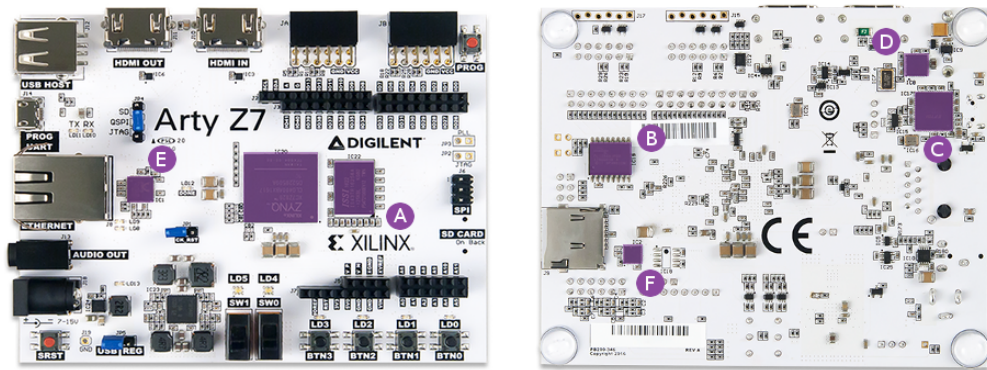
Các loại máy tính cơ bản thường được sử dụng rộng rãi trong đời sống hàng ngày và học tập phổ thông vì có chức năng đơn giản, dễ thao tác, với các phím bấm rõ ràng và màn hình nhỏ gọn hiển thị kết quả. Đối với những cá nhân làm việc trong các ngành liên quan đến tính toán, kỹ thuật thì máy tính khoa học hay máy tính đồ họa là sự lựa chọn tối ưu. Loại máy tính này có khả năng thực hiện các phép tính phức tạp như lượng giác, logarit, tích phân, tính toán ma trận và nhiều chức năng lập trình khác.

Nhờ sự đa dạng trong thiết kế và tính năng, các loại máy tính cầm tay ngày càng khẳng định vai trò quan trọng trong việc hỗ trợ học tập và nghiên cứu khoa học, không chỉ giúp tiết kiệm thời gian mà còn đảm bảo tính chính xác cao trong các bài toán mang tính chất phức tạp cao.

Trong đề tài này, nhóm chúng em sẽ vận dụng các kiến thức đã học trong môn thiết kế luận lý với HDL (CO1025) để tạo ra một máy tính đơn giản có thể thực hiện các phép tính cơ bản như cộng, trừ, nhân, chia và một số phép toán logic như AND, OR, XOR,... Dự án này không chỉ giúp nhóm hiểu rõ hơn về các nguyên lý thiết kế mạch logic số mà còn cung cấp cái nhìn chi tiết hơn về cách xây dựng một hệ thống tính toán từ các cổng logic cơ bản như AND, OR, NOT, XOR,... Hơn thế, việc sử dụng ngôn ngữ mô tả phần cứng (HDL) sẽ giúp nhóm thực hành thiết kế mạch một cách có hệ thống và hiệu quả, từ đó tạo ra một sản phẩm máy tính cơ bản nhưng hoàn chỉnh và khả thi trong môi trường thực tế.

1.2 Thiết bị sử dụng

Arty Z7-20 là một bo mạch dựa trên chip Xilinx Zynq-7000 (ZC702) SoC (System on Chip), kết hợp giữa một bộ xử lý ARM Cortex-A9 và một phần FPGA logic, cho phép bạn có thể sử dụng cả khả năng xử lý mạnh mẽ của vi xử lý và khả năng linh hoạt của FPGA trong một hệ thống duy nhất. Bo mạch này hỗ trợ các ứng dụng như xử lý tín hiệu số, điều khiển động cơ, xử lý ảnh, robot, và các ứng dụng hệ thống nhúng.



Hình 1.1: Hình ảnh mạch **Arty Z7-20**

Các thành phần chính của FPGA Arty Z7-20:

- **Xilinx Zynq-7000 SoC:**
 - **Bộ xử lý ARM Cortex-A9:** Cung cấp khả năng xử lý phần mềm mạnh mẽ với hai lõi ARM Cortex-A9 có thể chạy hệ điều hành Linux hoặc các hệ điều hành nhúng khác.

- **FPGA (Programmable Logic):** Phần FPGA trong Zynq-7000 có thể được lập trình để thực hiện các tác vụ logic phức tạp, chẳng hạn như xử lý tín hiệu, mã hóa/giải mã, điều khiển các thiết bị ngoại vi và nhiều ứng dụng khác.
- **Lập trình FPGA:**
 - Arty Z7-20 sử dụng công cụ Vivado Design Suite của Xilinx để lập trình và cấu hình phần FPGA. Bạn có thể sử dụng các ngôn ngữ mô tả phần cứng như Verilog hoặc VHDL để thiết kế các mạch logic và mô phỏng các hệ thống phần cứng phức tạp.
 - Cùng với phần xử lý ARM, phần FPGA có thể được sử dụng để thực hiện các tác vụ cần tốc độ xử lý cao và đồng thời với phần mềm điều khiển.
- **Các cổng kết nối đầu vào/đầu ra (I/O):**
 - **GPIO Pins:** Arty Z7-20 cung cấp rất nhiều chân GPIO cho phép bạn kết nối các cảm biến, thiết bị ngoại vi, đèn LED, và các thiết bị khác. Đây là các chân có thể lập trình được, bạn có thể dùng chúng để giao tiếp với các hệ thống bên ngoài.
 - **USB, HDMI, Audio, Ethernet:** Các giao diện này giúp kết nối Arty Z7-20 với các thiết bị ngoại vi như màn hình, âm thanh, hoặc kết nối mạng, phục vụ cho các ứng dụng phức tạp hơn.
 - **Mạch mở rộng (Extension Board):** Như đã đề cập trước đó, Arty Z7-20 có thể kết nối với các bảng mạch mở rộng (extension board) để bổ sung các cổng kết nối và cảm biến thêm cho các ứng dụng của bạn.
- **Bộ nhớ và khả năng lưu trữ:**

- **On-board Flash Memory:** Arty Z7-20 có bộ nhớ flash tích hợp để lưu trữ cấu hình FPGA và phần mềm cho hệ thống.
- **SD Card Slot:** Bo mạch này còn cung cấp khe cắm thẻ SD, cho phép lưu trữ dữ liệu hoặc chương trình hệ điều hành nhúng.
- **Cung cấp nguồn điện:**
 - Bo mạch yêu cầu nguồn điện ngoài và hỗ trợ một số phương thức cấp nguồn thông qua cổng micro-USB hoặc J12 Power Connector. Điều này giúp cung cấp năng lượng cho cả phần xử lý ARM và phần FPGA.

Ưu điểm của FPGA Arty Z7-20:

- **Tính linh hoạt:**
 - FPGA Arty Z7-20 có khả năng cấu hình lại phần cứng, giúp bạn tùy chỉnh hệ thống theo yêu cầu cụ thể của dự án mà không phải thay đổi phần cứng vật lý.
- **Hiệu suất cao:**
 - Với khả năng thực hiện các phép toán song song và tối ưu hóa các tác vụ phần cứng, Arty Z7-20 rất thích hợp cho các ứng dụng yêu cầu hiệu suất cao, chẳng hạn như xử lý tín hiệu số (DSP), mã hóa/giải mã, hoặc điều khiển phức tạp.
- **Tích hợp phần cứng và phần mềm:**
 - Arty Z7-20 kết hợp giữa phần cứng FPGA và phần mềm ARM, giúp bạn tận dụng lợi thế của cả hai: phần mềm chạy trên ARM Cortex-A9 có thể xử lý các tác vụ quản lý hệ thống và giao tiếp với người dùng, trong khi phần FPGA xử lý các tác vụ yêu cầu tốc độ cao và song song.

Các ứng dụng của FPGA Arty Z7-20:

- **Hệ thống điều khiển nhúng:**

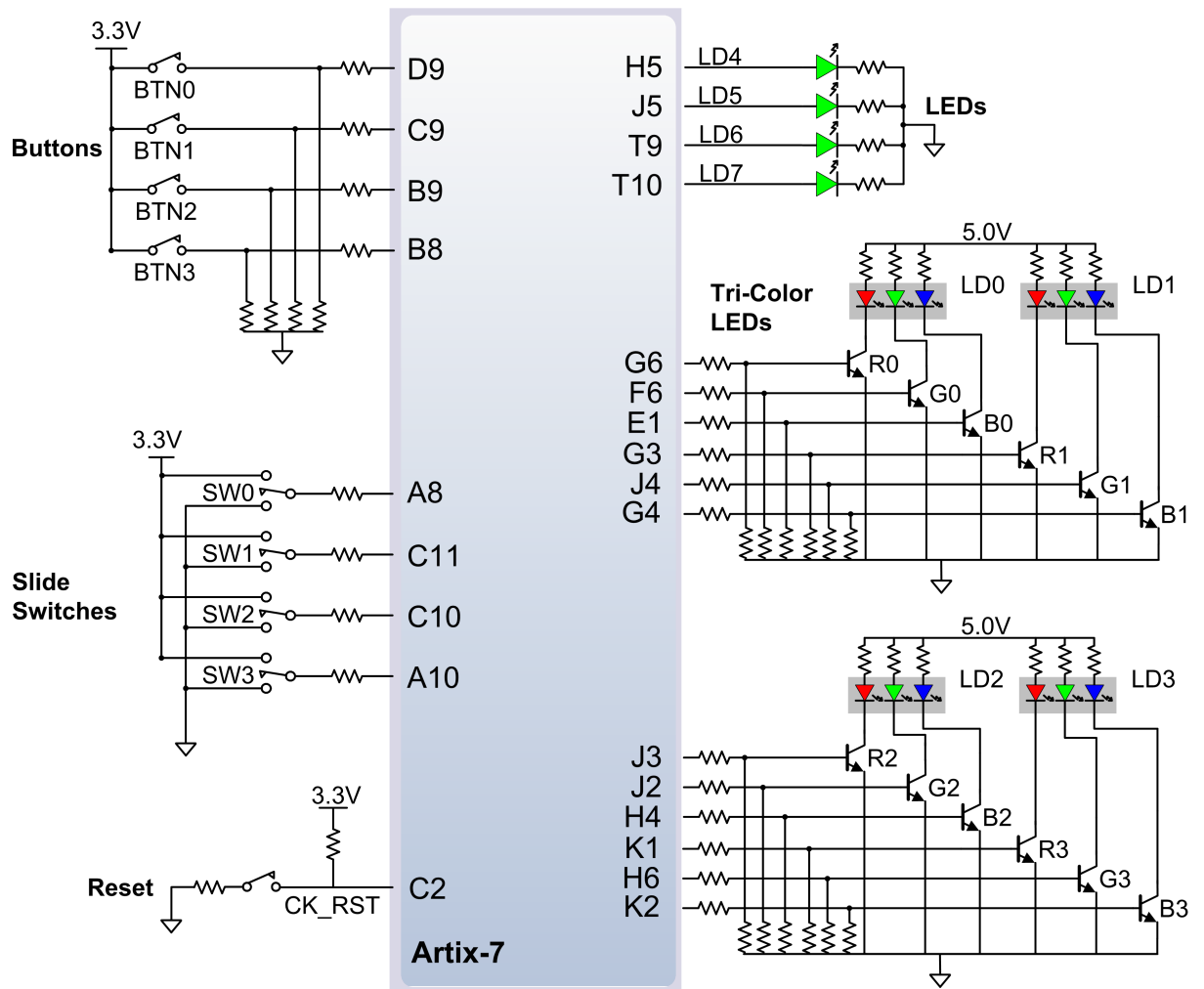
- Arty Z7-20 có thể được sử dụng để thiết kế các hệ thống điều khiển nhúng, chẳng hạn như trong các robot, xe tự lái, hoặc các hệ thống điều khiển tự động khác.

- **Xử lý tín hiệu số (DSP):**

- FPGA rất mạnh trong việc xử lý các tín hiệu số, như âm thanh, hình ảnh hoặc dữ liệu từ cảm biến, và Arty Z7-20 cung cấp một môi trường lý tưởng để phát triển các ứng dụng này.

- **Máy tính bấm tay và hệ thống tính toán:**

- Với khả năng lập trình linh hoạt, Arty Z7-20 có thể được sử dụng trong các dự án máy tính bấm tay, máy tính đơn giản, hoặc hệ thống tính toán tùy chỉnh, như trong dự án.



Hình 1.2: Cấu trúc sơ đồ GPIO mạch ARTY Z7-20

1.3 Công cụ sử dụng

Trong đề tài này nhóm đã sử dụng phần mềm phát triển và mô phỏng thiết kế FPGA Vivado Design Suite.

Vivado là công cụ chủ yếu dùng để thiết kế phần cứng FPGA, hỗ trợ một loạt các tính năng từ mô phỏng, thiết kế logic, kiểm tra, cho đến tối ưu hóa tài nguyên hệ thống. Được phát triển để hỗ trợ các dòng FPGA hiện đại của Xilinx, Vivado thay thế công cụ ISE cũ và mang đến một môi trường phát triển mạnh mẽ và hiệu quả hơn.



Các tính năng chính của Vivado:

- **Thiết kế logic và mô phỏng:**
 - Vivado cung cấp một giao diện đồ họa trực quan giúp lập trình viên dễ dàng thiết kế các hệ thống FPGA bằng ngôn ngữ mô tả phần cứng như Verilog, VHDL, hoặc thông qua Block Diagram (Sơ đồ khối).
 - Hệ thống mô phỏng trong Vivado hỗ trợ mô phỏng phần cứng để kiểm tra thiết kế trước khi chuyển qua bước triển khai thực tế. Vivado tích hợp công cụ ModelSim cho phép mô phỏng các tín hiệu logic và phát hiện lỗi trong giai đoạn thiết kế.
- **Tối ưu hóa thiết kế:**

- Vivado cung cấp một giao diện đồ họa trực quan giúp lập trình viên dễ dàng thiết kế các hệ thống FPGA bằng ngôn ngữ mô tả phần cứng như Verilog, VHDL, hoặc thông qua Block Diagram (Sơ đồ khối).
- Hệ thống mô phỏng trong Vivado hỗ trợ mô phỏng phần cứng để kiểm tra thiết kế trước khi chuyển qua bước triển khai thực tế. Vivado tích hợp công cụ ModelSim cho phép mô phỏng các tín hiệu logic và phát hiện lỗi trong giai đoạn thiết kế.
- **Tạo bitstream:**
 - Sau khi thiết kế được hoàn thành và kiểm tra qua mô phỏng, Vivado có thể tạo bitstream file – một tập tin nhị phân được sử dụng để lập trình FPGA.
 - Bitstream này sẽ được nạp vào FPGA để thực hiện các tác vụ mà bạn đã lập trình.
- **Hỗ trợ cho hệ thống SoC (System on Chip):**
 - Vivado không chỉ hỗ trợ thiết kế phần cứng FPGA thuần túy mà còn hỗ trợ thiết kế SoC, nơi FPGA và vi xử lý ARM Cortex-A9 (hoặc các vi xử lý khác của Xilinx) được kết hợp trong cùng một hệ thống.
 - Bạn có thể sử dụng Vivado để thiết kế các hệ thống phức tạp, tích hợp phần cứng và phần mềm, và đồng thời tối ưu hóa các kết nối giữa phần cứng FPGA và phần mềm chạy trên vi xử lý.
- **Thiết kế theo dòng thời gian (Timing):**
 - Vivado cung cấp các công cụ phân tích thời gian mạnh mẽ, giúp đảm bảo rằng hệ thống FPGA hoạt động trong các yêu cầu về thời gian (timing constraints). Bạn có thể xác định các điều kiện thời gian của tín hiệu và xem xét cách thức hoạt động của các tín hiệu trong hệ thống.

- **Hỗ trợ cho IP (Intellectual Property) Cores:**

- Vivado cung cấp một thư viện phong phú các IP cores đã được tối ưu hóa cho các FPGA của Xilinx. Các IP này có thể là các chức năng đã được chuẩn hóa và tối ưu như bộ mã hóa, giải mã, bộ xử lý tín hiệu số (DSP), giao thức mạng, bộ nhớ, v.v.
- Vivado còn hỗ trợ tạo ra các IP tùy chỉnh, cho phép bạn xây dựng các mô-đun phần cứng riêng biệt và tái sử dụng chúng trong các dự án khác.

- **Chế độ thiết kế hệ thống (System Generator):**

- Vivado hỗ trợ chế độ thiết kế hệ thống (System Generator) cho các ứng dụng DSP, giúp bạn tạo ra các mô-đun tín hiệu số phức tạp mà không cần phải viết mã thủ công.

- **Giao diện người dùng (GUI) và dòng lệnh (Tcl):**

- Vivado cung cấp cả giao diện người dùng đồ họa (GUI) để sử dụng và giao diện dòng lệnh (Tcl scripting) giúp tự động hóa các tác vụ và tích hợp vào quy trình phát triển của bạn.

1.4 Chức năng sản phẩm

Mục tiêu thiết kế:

Mục tiêu của dự án là phát triển một máy tính với khả năng thực hiện các phép toán đại số cơ bản (cộng, trừ, nhân, chia, chia lấy dư) và phép toán logic AND của hai số nhị phân. Máy tính này sẽ có các tính năng hỗ trợ người dùng nhập vào tối thiểu hai giá trị (0 và 1) và lựa chọn phép toán qua các nút bấm (buttons) và công tắc (switches). Kết quả của các phép toán sẽ được hiển thị qua hệ thống LED 7 đoạn (dưới dạng số thập phân), với các chức năng bổ sung như ghi nhớ kết quả trước đó để tiếp tục tính toán trong các phép toán sau.

Chi tiết các yêu cầu chức năng:

Máy tính sẽ thực hiện phép toán giữa 2 số nhị phân 16 bit. Kết quả trả về sẽ là một số thập phân, và được hiển thị trên 4 LED 7 đoạn.

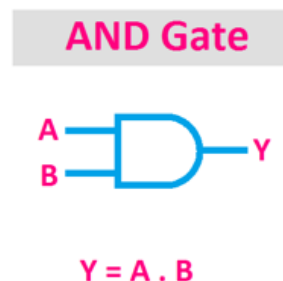
- **Lưu ý của phép toán đại số:**

- Máy tính hỗ trợ 5 phép chia đại số: **cộng, trừ, nhân, chia và chia lấy dư.**
- Khi người dùng nhập 1 bit mới thì hệ thống sẽ dịch các bit đã có sang trái và gán bit vừa nhập vào bit có trọng số lớn nhất.
- Trong trường hợp người dùng nhập không đủ số lượng bit quy định thì hệ thống sẽ mặc định gán giá trị 0 cho các bit thiếu.
- Trong trường hợp kết quả là số âm, hệ thống sẽ tự động chuyển đổi kết quả về giá trị tuyệt đối và hiển thị trên LED 7 đoạn.
- **Chia (Division):** Trong trường hợp số chia bằng 0, kết quả sẽ hiển thị là "9999" để báo lỗi.

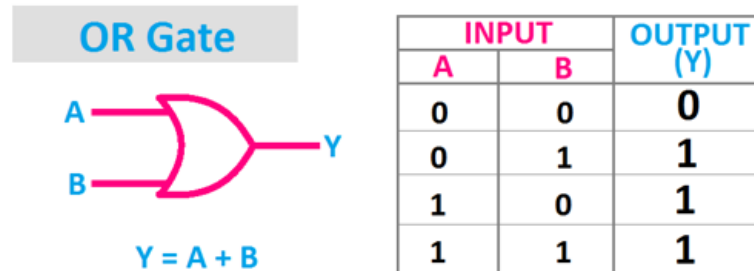
- **Các phép toán logic:**

- **Phép toán logic AND:** Máy tính sẽ thực hiện phép toán logic AND giữa hai giá trị nhị phân 16 bit. Kết quả sẽ được hiển thị dưới dạng nhị phân.

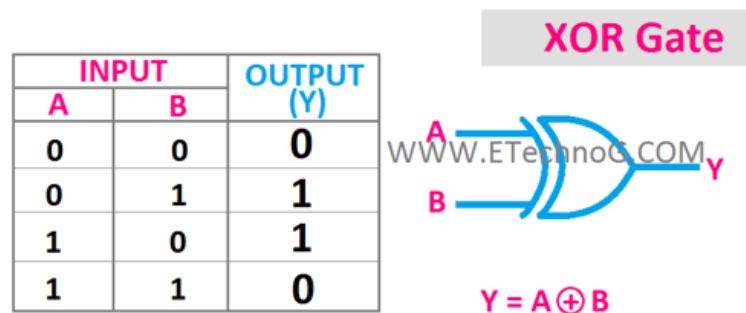
INPUT		OUTPUT (Y)
A	B	
0	0	0
0	1	0
1	0	0
1	1	1



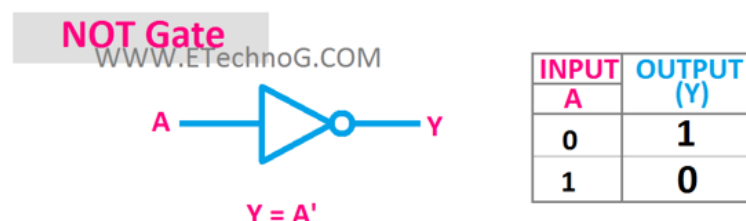
- **Phép toán logic OR:** Phép toán OR thực hiện so sánh giữa hai giá trị nhị phân 16 bit và trả về một giá trị nhị phân 16 bit. Kết quả của phép toán OR sẽ là 1 nếu ít nhất một trong hai bit tương ứng là 1. Ngược lại, kết quả sẽ là 0.



- **Phép toán logic XOR:** Phép toán XOR (Exclusive OR) cũng so sánh hai giá trị nhị phân 16 bit, nhưng kết quả sẽ là 1 chỉ khi hai bit so sánh khác nhau (một là 0, một là 1). Nếu cả hai bit giống nhau (0,0 hoặc 1,1), kết quả sẽ là 0.



- **Phép toán logic NOT:** Phép toán NOT sẽ áp dụng cho một giá trị nhị phân 16 bit và đảo ngược mỗi bit. Kết quả sẽ là một giá trị nhị phân 16 bit, trong đó các bit 1 trở thành 0 và ngược lại.



2

Lý thuyết cơ bản

Máy tính cơ bản là một hệ thống số được thiết kế để thực hiện các phép toán số học và logic đơn giản. Thiết bị này được xây dựng dựa trên các khối logic cơ bản trong hệ thống số và hoạt động theo nguyên tắc của bộ xử lý cơ bản. Sản phẩm được triển khai trên nền FPGA, sử dụng các tài nguyên logic và bộ nhớ tích hợp để thực hiện tính toán. FPGA (Field-Programmable Gate Array) cho phép thiết kế các mạch logic có thể lập trình lại, mang lại tính linh hoạt cao và khả năng tối ưu hóa phần cứng cho các ứng dụng đặc thù, như máy tính cơ bản trong dự án này.

Các phép toán này được thực hiện trên các số nhị phân 16 bit, với các kết quả đầu ra cũng là các giá trị nhị phân, sau đó được chuyển đổi thành giá trị thập phân để hiển thị trên các LED 7 đoạn. Hệ thống này không chỉ bao gồm phần xử lý dữ liệu mà còn tích hợp khả năng giao tiếp với người dùng thông qua các công tắc (switches) và nút bấm (buttons), cho phép người dùng chọn các phép toán và nhập các giá trị đầu vào.

Nội dung

2.1	Đặc tả hệ thống	19
2.1.1	Nhập dữ liệu từ người dùng:	19
2.1.2	Xử lý dữ liệu bên trong hệ thống	20
2.1.3	Hiển thị kết quả	20
2.2	Những tính năng được ứng dụng trong thực tế	20

2.1 Đặc tả hệ thống

2.1.1 Nhập dữ liệu từ người dùng:

Hệ thống nút/nhấn Button:

Các nút bấm sẽ được sử dụng để nhập các giá trị nhị phân và lưu vào các biến trong hệ thống. Sau khi nhập xong, người dùng cần xác nhận giá trị nhập bằng cách nhấn một nút xác nhận.

- Button 0 (BTN0) và Button 1 (BTN1): Nhập các bit nhị phân (0 hoặc 1).
- Button 2 (BTN2) và Button 3 (BTN3): Lưu các giá trị nhập vào hai biến a và b để làm các giá trị đầu vào cho phép toán.

Switch chức năng:

Các công tắc (switchs) được sử dụng để thay đổi chức năng của máy tính đơn giản, hoặc để thay đổi trạng thái của input (trong trường hợp người dùng có nhu cầu sử dụng lại kết quả đã tính trước đó). Các switch này được sử dụng trên Extension Board for Arty Z7-20.

- Switch 0 (SW0): Thực hiện phép toán cộng.
- Switch 1 (SW1): Thực hiện phép toán trừ.
- Switch 2 (SW2): Thực hiện phép toán nhân.
- Switch 3 (SW3): Thực hiện phép toán chia.
- Switch 4 (SW4): Thực hiện phép toán chia có dư.
- Switch 5 (SW5): Thực hiện phép toán AND.
- Switch 6 (SW6): Lưu kết quả phép toán gần nhất vào biến ans.
- Switch Reset (trên Arty Z7-20): Thực hiện lệnh reset để bắt đầu lại một phép toán mới.

2.1.2 Xử lý dữ liệu bên trong hệ thống

Các hàm xử lý:

Khi người dùng nhập dữ liệu và chọn phép toán, hệ thống sẽ nhận các input từ các nút bấm và công tắc. Sau đó, tùy thuộc vào phép toán được chọn, hệ thống sẽ gọi các hàm xử lý tương ứng để thực hiện phép toán.

- Ví dụ, nếu phép toán hiện tại là phép cộng, hệ thống sẽ gọi hàm thực hiện phép cộng và xử lý các giá trị đầu vào a và b.
- Các phép toán khác như trừ, nhân, chia, và các phép toán logic sẽ được xử lý theo cách tương tự.

2.1.3 Hiển thị kết quả

Kết quả của phép toán sẽ được hiển thị trên hệ thống LED 7 đoạn. Mỗi kết quả tính toán sẽ được chuyển đổi từ giá trị nhị phân sang giá trị thập phân và hiển thị tối đa 4 chữ số thập phân trên LED 7 đoạn.

2.2 Những tính năng được ứng dụng trong thực tế

Trong học tập:

- Hiểu rõ nguyên lý hoạt động của các khối cơ bản trong máy tính.
- Rèn luyện kỹ năng thiết kế mạch số và lập trình HDL (Hardware Description Language).
- Giúp sinh viên và kỹ sư mới hiểu và nắm vững cách xây dựng và vận hành một máy tính cơ bản từ các mạch logic cơ bản.
- Cung cấp nền tảng vững chắc cho các môn học về điện tử số, vi xử lý, và thiết kế vi mạch LSI (Large Scale Integration).

- Thực hành thiết kế mạch trên FPGA giúp sinh viên có trải nghiệm thực tế với công nghệ phần cứng, nâng cao khả năng giải quyết vấn đề và sáng tạo.

Trong thực tế:

- Có thể được mở rộng thành các máy tính bỏ túi hoặc hệ thống tính toán đơn giản.
 - Ví dụ, có thể tạo ra một máy tính bỏ túi với các chức năng cơ bản như cộng, trừ, nhân, chia, và các phép toán logic số nhị phân.
 - Cũng có thể phát triển thành các ứng dụng hỗ trợ tính toán trong các thiết bị di động hoặc máy tính cầm tay.
- Là bước đầu để phát triển các hệ thống phức tạp hơn như vi xử lý hoặc SoC (System on Chip).
 - Dự án này có thể được coi là nền tảng cho việc phát triển các vi xử lý đơn giản, nơi mà mỗi chức năng như cộng, trừ hay nhân được thực hiện trên các khối xử lý riêng biệt.
 - Sự kết hợp của các khối chức năng này có thể mở rộng để xây dựng một SoC với khả năng xử lý mạnh mẽ hơn, bao gồm cả các tính năng như xử lý tín hiệu số, điều khiển, và giao tiếp với các thiết bị ngoại vi.
- Cung cấp cái nhìn sâu sắc vào cách thức các hệ thống tính toán có thể được tối ưu hóa và tích hợp vào các ứng dụng thực tế.

Trong công nghiệp:

- Ứng dụng trong các hệ thống điều khiển đơn giản hoặc hệ thống nhúng.
 - Ví dụ, có thể sử dụng cho các hệ thống điều khiển tự động trong các thiết bị gia dụng, máy móc công nghiệp hoặc thiết bị y tế.

- Các hệ thống nhúng có thể sử dụng các tính toán đơn giản cho các ứng dụng như đo lường, kiểm tra chất lượng sản phẩm, hoặc xử lý tín hiệu cơ bản.
- Cung cấp giải pháp cho các hệ thống nhúng thời gian thực, nơi yêu cầu tính toán chính xác và nhanh chóng, chẳng hạn như trong các thiết bị đo lường, điều khiển máy móc, hoặc hệ thống an toàn.
- Dùng trong các ứng dụng IoT (Internet of Things), nơi các thiết bị cần thực hiện các phép toán cơ bản hoặc xử lý dữ liệu trực tiếp trên thiết bị mà không cần kết nối với máy chủ.
- Tạo nền tảng cho các sản phẩm điện tử tiêu dùng có yêu cầu tính toán nhẹ nhưng hiệu quả, như đồng hồ thông minh, các thiết bị wearable, hoặc các thiết bị y tế thông minh.

3

Thiết kế

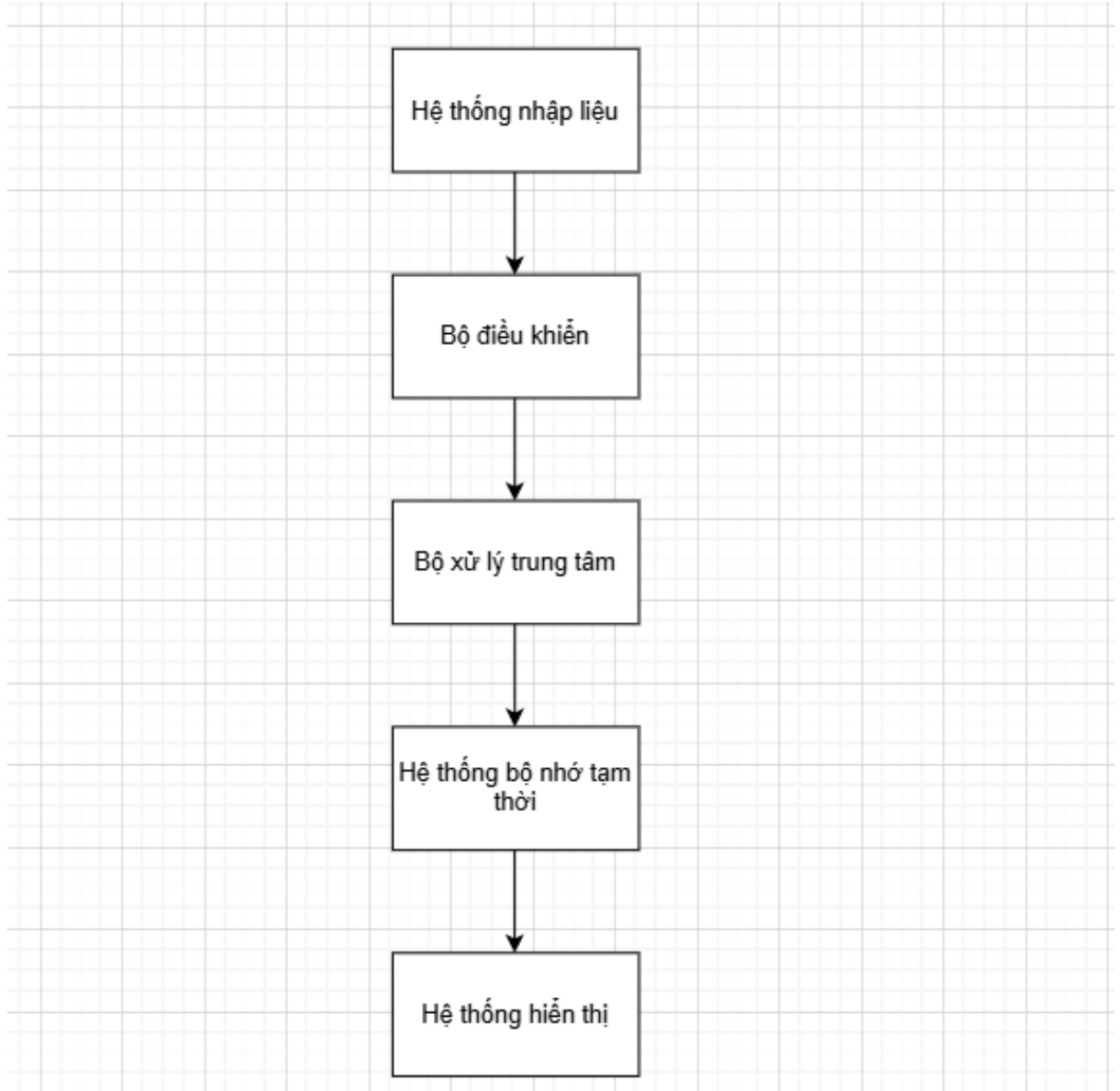
Trong chương này, chúng tôi sẽ mô tả chi tiết hơn về việc thiết kế hệ thống trên. Mục tiêu của thiết kế là xây dựng một hệ thống tính toán cơ bản có khả năng thực hiện các phép toán số học và logic đơn giản, đồng thời hỗ trợ giao tiếp với người dùng thông qua các nút bấm và công tắc, và hiển thị kết quả trên các LED 7 đoạn.

Nội dung

3.1	Sơ đồ khối	24
3.1.1	Bộ xử lý trung tâm (ALU - Arithmetic Logic Unit): .	25
3.1.2	Bộ điều khiển (Control Unit)	25
3.1.3	Hệ thống nhập liệu:	26
3.1.4	Hệ thống hiển thị:	26
3.1.5	Bộ nhớ tạm thời (Register):	26

3.1 Sơ đồ khối

Các khối cơ bản của máy tính cơ bản:



Hình 3.1: Sơ đồ khối hệ thống

3.1.1 Bộ xử lý trung tâm (ALU - Arithmetic Logic Unit):

Chức năng: Bộ xử lý trung tâm (ALU) chịu trách nhiệm thực hiện tất cả các phép toán số học và logic trong hệ thống. Nó nhận dữ liệu đầu vào từ bộ nhớ và các phần tử khác trong hệ thống, thực hiện các phép toán theo yêu cầu và trả kết quả về bộ nhớ hoặc các thiết bị hiển thị.

Các phép toán số học:

- **Cộng:** Thực hiện phép cộng hai số nhị phân 16 bit.
- **Trừ:** Thực hiện phép trừ hai số nhị phân, với xử lý đặc biệt khi kết quả là giá trị âm (hệ thống sẽ tự động lấy trị tuyệt đối).
- **Nhân:** Thực hiện phép nhân hai số, với kết quả đầu ra là 32 bit.
- **Chia:** Thực hiện phép chia và chia có dư, với kết quả đầu ra 32 bit.
- **Modulo:** Thực hiện phép chia lấy dư, kết quả là phần dư sau khi chia.

Các phép toán logic:

- **AND:** Các phép toán logic cơ bản được thực hiện giữa các giá trị nhị phân, hỗ trợ các chức năng kiểm tra điều kiện và xử lý tín hiệu.
- Có thể bổ sung thêm các phép toán logic khác như **OR, NOT, XOR,...**

3.1.2 Bộ điều khiển (Control Unit)

Chức năng: Bộ điều khiển là phần quan trọng của hệ thống, có nhiệm vụ điều phối và quản lý các hoạt động của các khối trong hệ thống. Nó nhận tín hiệu từ các công tắc (switches) và nút bấm (buttons), phân tích lệnh của người dùng, và chuyển sang chế độ thực hiện tương ứng (như thực hiện phép toán cộng, trừ, nhân, chia,...).

Điều phối chế độ hoạt động: Bộ điều khiển sẽ quyết định hệ thống có đang ở trạng thái tính toán hay nhập dữ liệu, từ đó sẽ điều phối hoạt động của ALU, bộ nhớ tạm thời (register) và các thiết bị hiển thị. Bộ điều khiển cũng có thể sử

dùng tín hiệu reset để đưa hệ thống về trạng thái ban đầu khi người dùng yêu cầu.

3.1.3 Hệ thống nhập liệu:

Chức năng: Hệ thống nhập liệu bao gồm các nút bấm và công tắc, cho phép người dùng nhập các giá trị nhị phân, chọn phép toán cần thực hiện và điều khiển hệ thống.

Nút bấm: Các nút bấm như BTN0 và BTN1 được dùng để nhập các bit 0 và 1. Các nút khác như BTN2 và BTN3 dùng để lưu giá trị vào các biến a và b.

Công tắc: Các công tắc (SW0 đến SW6) được dùng để lựa chọn phép toán (cộng, trừ, nhân, chia, chia lấy dư và AND), lưu kết quả vào ans và reset hệ thống.

3.1.4 Hệ thống hiển thị:

Chức năng: Hệ thống hiển thị có nhiệm vụ hiển thị các giá trị đầu vào, kết quả phép toán và các tín hiệu trạng thái khác của hệ thống.

LED 7 đoạn: Được sử dụng để hiển thị kết quả của phép toán dưới dạng thập phân. Mỗi giá trị kết quả sẽ được chuyển đổi từ nhị phân sang thập phân và hiển thị lên tối đa 4 chữ số trên màn hình LED 7 đoạn. Hệ thống LED giúp người dùng dễ dàng theo dõi kết quả tính toán trong thời gian thực.

3.1.5 Bộ nhớ tạm thời (Register):

Chức năng: Bộ nhớ tạm thời (register) lưu trữ kết quả của phép tính trước đó để sử dụng trong các phép tính tiếp theo. Khi người dùng yêu cầu, bộ nhớ này cũng có thể lưu trữ kết quả phép toán gần nhất vào ans.

Sử dụng:

- Các giá trị a và b sẽ được lưu vào register khi người dùng nhập chúng thông qua các nút bấm.

- Kết quả của các phép toán sẽ được lưu vào register để có thể sử dụng cho các phép toán tiếp theo, điều này hỗ trợ thực hiện các tính toán phụ thuộc vào kết quả trước đó.

4

Hiện thực

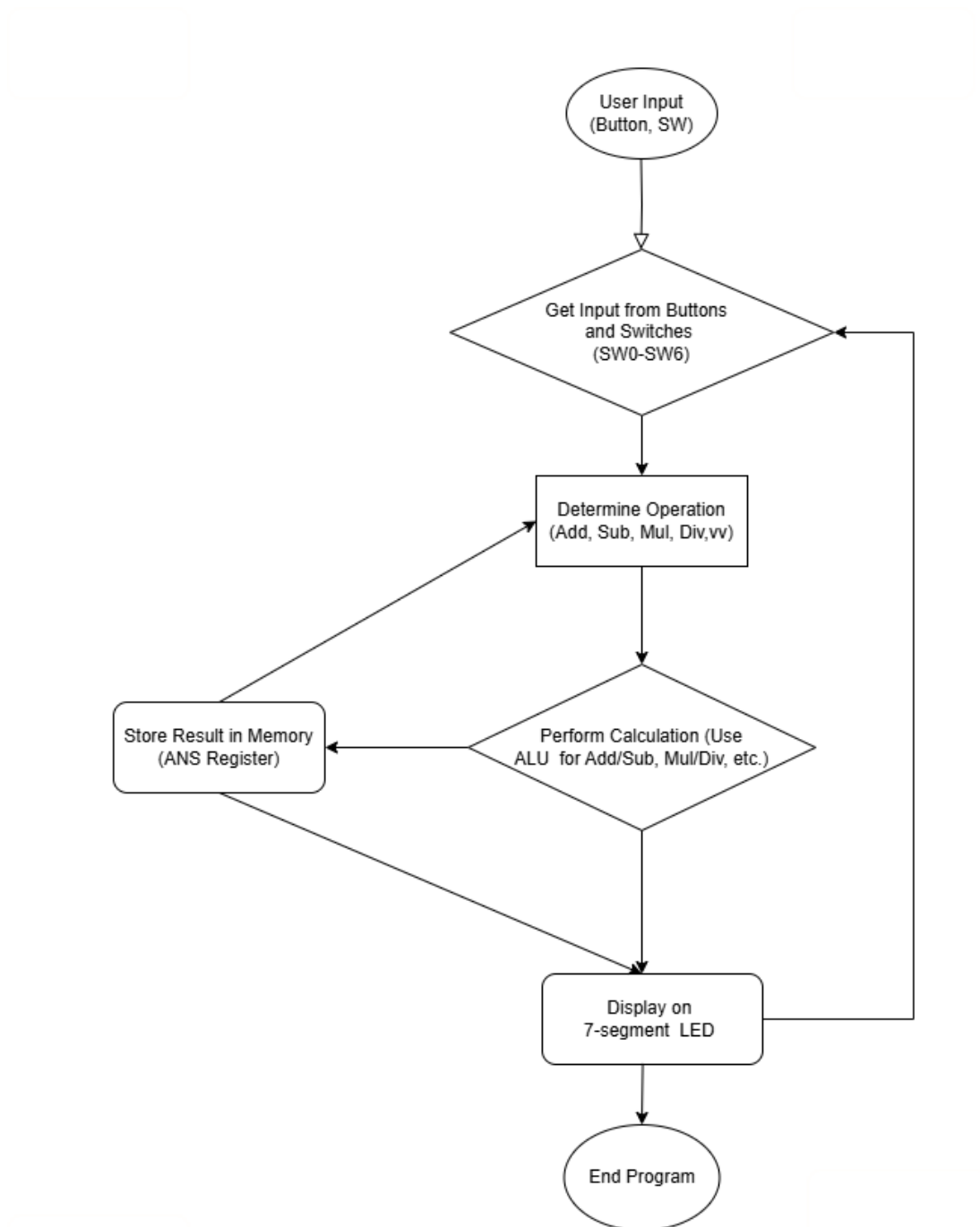
Sau khi đã hoàn tất quá trình thiết kế lý thuyết và xác định các khối cơ bản của hệ thống, chúng tôi sẽ tiến hành triển khai hệ thống trên phần cứng FPGA. Việc hiện thực hệ thống sẽ được chia thành nhiều giai đoạn, từ việc mô tả các mạch logic trong ngôn ngữ mô tả phần cứng (HDL) đến việc kiểm tra và xác nhận chức năng của hệ thống trên nền tảng FPGA.

Nội dung

4.1	Sơ đồ khối	29
-----	----------------------	----

4.1 Sơ đồ khối

Sơ đồ khối hiện thực:



Hình 4.1: Sơ đồ khối hiện thực

Người dùng sẽ bắt đầu bằng cách sử dụng các công tắc (switch) trên mạch để nhập các giá trị và chọn phép toán cần thực hiện. Các giá trị đầu vào được biểu thị qua các công tắc này, bao gồm các số và phép toán mong muốn. Sau khi hoàn tất việc nhập dữ liệu, hệ thống sẽ chuyển qua bộ ALU (Arithmetic Logic Unit) để thực hiện tính toán theo yêu cầu.

Kết quả phép toán sau khi được tính toán có thể được lưu vào bộ nhớ để sử dụng cho các phép toán tiếp theo hoặc hiển thị trực tiếp ra màn hình LED 7 đoạn để người dùng xem. Hệ thống hỗ trợ việc lưu trữ kết quả cuối cùng vào một biến nhớ tạm thời (register), giúp người dùng có thể sử dụng kết quả này trong các phép toán tiếp theo mà không cần nhập lại các giá trị ban đầu. Nếu người dùng yêu cầu, kết quả này có thể được lưu vào biến Ans để tái sử dụng cho các phép toán phụ thuộc vào kết quả trước đó.

Sau khi hiển thị kết quả, nếu người dùng muốn tiếp tục thực hiện các phép toán mới, họ có thể quay lại bước nhập giá trị đầu vào. Khi đó, hệ thống sẽ quay lại trạng thái chờ nhập giá trị mới từ các công tắc và nút bấm. Người dùng có thể thay đổi các giá trị đầu vào hoặc chọn phép toán khác thông qua công tắc. Quá trình này sẽ tiếp tục cho đến khi người dùng chọn lệnh reset hoặc dừng lại.

Nếu không, chương trình sẽ kết thúc và dừng lại. Để kết thúc chương trình, người dùng có thể sử dụng công tắc reset (SW0 hoặc một công tắc reset riêng biệt) để đưa hệ thống về trạng thái ban đầu, nơi tất cả các biến và bộ nhớ sẽ được xóa, chuẩn bị cho một lần nhập mới. Sau khi reset, hệ thống sẽ trở lại chế độ sẵn sàng để người dùng nhập dữ liệu hoặc chọn phép toán mới.

5

Kết luận

Trong quá trình thực hiện đồ án thiết kế luận lý, nhóm đã thiết kế một máy tính đơn giản sử dụng FPGA Arty Z7-20. Qua đó, nhóm đã hiểu rõ hơn về cách thức hoạt động của các mạch logic số trong FPGA và ứng dụng kiến thức này để xây dựng một hệ thống tính toán cơ bản. Ngoài ra, nhóm cũng học cách viết một chương trình Verilog hoàn chỉnh cho hệ thống. Quá trình thiết kế bao gồm các bước từ xác định yêu cầu phần cứng, thiết kế các mạch logic, triển khai lên phần cứng, kiểm tra kết quả và điều chỉnh sao cho hệ thống đạt yêu cầu đề ra.

Với khả năng xử lý mạnh mẽ và khả năng tái cấu trúc mạch logic linh hoạt của mạch FPGA Arty Z7-20 đã cho phép xây dựng một hệ thống máy tính có thể thực hiện các phép toán đơn giản như cộng, trừ, nhân, chia và quản lý các thao tác nhập/xuất cơ bản. Việc áp dụng FPGA trong thiết kế máy tính không chỉ giúp tiết kiệm chi phí mà còn mang lại sự linh hoạt trong việc tối ưu hóa các thành phần của hệ thống theo yêu cầu cụ thể.

Kết quả đạt được từ đề tài đã chứng minh rằng FPGA là một công cụ phần cứng khá lý tưởng để nghiên cứu và phát triển các hệ thống tính toán với các tính năng cơ bản, và có thể mở rộng ra các ứng dụng phức tạp hơn trong tương lai nhờ các mạch mở rộng. Tuy nhiên, vẫn còn những thách thức cần phải khắc phục như việc tối ưu hóa hiệu suất và việc nâng cao khả năng tương tác với các thiết bị ngoại vi.

Cuối cùng, nghiên cứu này không chỉ giúp chúng em phát triển kỹ năng thiết kế hệ thống trên nền tảng FPGA mà còn cung cấp một nền tảng vững chắc cho những nghiên cứu tiếp theo trong việc ứng dụng FPGA trong các hệ thống tính toán thực tế.

6

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] *Hướng dẫn tài liệu vivado* <https://www.xilinx.com/support/installer/installer-info-general.html>
- [2] Slide course Logic Design Project
- [3] Slide course Digital Systems
- [4] Tài liệu mạch Arty-A7, truy cập từ: <https://digilent.com/reference/programmable-logic/arty-a7/reference-manual>
- [5] Developerdia, *Low-Power Wide-Area Network*, [Online], truy cập từ <https://devopedia.org/low-power-wide-area-network>
- [6] Ayse Muhammetoglu, Yalçın Albayrak, Mustafa Bolbol, Simge Enderoglu, Habib Muhammetoglu, "Detection and Assessment of Post Meter Leakages in Public Places Using Smart Water Metering"
- [7] Ansuman Adhikary, Xingqin Lin, Y.-P. Eric Wan, "Performance Evaluation of NB-IoT Coverage"
- [8] Victor Gayuh Utomo, Nurtriana Hidayati, Agusta Praba Ristadi Pinem, "Evaluation Metrics of Water Meter Reading with Optical Character Recognition"