

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA ĐIỆN TỬ VIỄN THÔNG  
BỘ MÔN MÁY TÍNH – HỆ THỐNG NHÚNG**



**PHAN TÂN**

**Đề tài:**

**NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG  
MINING ASIC TRÊN NỀN TẢNG FPGA**

**Chuyên ngành Máy Tính - Hệ Thống Nhúng**

**TP. Hồ Chí Minh, tháng 7 năm 2023**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA ĐIỆN TỬ - VIỄN THÔNG  
BỘ MÔN MÁY TÍNH - HỆ THỐNG NHÚNG**



**PHAN TÂN  
MSSV: 19200473**

**Đề tài:**

**NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG  
MINING ASIC TRÊN NỀN TẢNG FPGA**

**RESEARCH AND BUILD MINING ASIC SYSTEM ON FPGA**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN  
NGÀNH KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG  
CHUYÊN NGÀNH MÁY TÍNH - HỆ THỐNG NHÚNG**

**NGƯỜI HƯỚNG DẪN KHOA HỌC  
GV. Trần Tuấn Kiệt**

**TP. Hồ Chí Minh, tháng 7 năm 2023**

## LỜI CẢM ƠN

Trước tiên em xin gửi lời cảm ơn sâu sắc nhất đến thầy Trần Tuấn Kiệt. Thầy đã tận tâm, tận lực hướng dẫn, định hướng em nghiên cứu khoa học cho khóa luận này, đồng thời cũng đã cung cấp nhiều tài liệu và tạo điều kiện thuận lợi trong suốt quá trình học tập và nghiên cứu để em có thể hoàn thành khóa luận này.

Em xin được gửi lời cảm ơn đến các Thầy, Cô trong Khoa Điện Tử - Viễn Thông trường Đại học Khoa học Tự Nhiên đã nhiệt tình giảng dạy và truyền đạt những kiến thức, kinh nghiệm quý giá trong suốt thời gian em học tập tại trường.

Cảm ơn gia đình, bạn bè đã quan tâm và động viên giúp em có nghị lực phấn đấu để hoàn thành tốt luận văn này.

Do kiến thức và thời gian có hạn nên luận văn chắc chắn không tránh khỏi những thiếu sót nhất định. Một lần nữa em xin gửi lời cảm ơn chân thành và sâu sắc.

## TÓM TẮT LUẬN VĂN

Trong thời đại kỹ thuật số phát triển nhanh chóng, tiền điện tử đã nổi lên như một sự cách mạng trong lĩnh vực tài chính. Tiền điện tử không chỉ đại diện cho một hình thức thanh toán tiện lợi, mà còn mang trong mình tiềm năng thay đổi toàn bộ cách thức giao dịch, quản lý tài sản và trao đổi giá trị.

Tiềm năng của việc khai thác tiền điện tử là rất lớn, quá trình khai thác tạo ra một cơ chế tự động để phát hành tiền điện tử mới và phân phối nó cho các thành viên trong mạng lưới. Điều này tạo ra một hệ thống kinh tế phi tập trung, không phụ thuộc vào các cơ quan tài chính trung gian và ngân hàng truyền thống. Nhờ vậy, tiền điện tử mang lại sự tự chủ và khả năng tiếp cận tài chính cho mọi người trên toàn cầu, bất kể vị trí địa lý hay tình trạng tài chính cá nhân. Ngoài ra, việc khai thác tiền điện tử cung cấp một cơ chế an toàn và minh bạch cho việc xác nhận giao dịch trong mạng lưới. Các nguyên tắc mã hóa và công nghệ blockchain được áp dụng trong quá trình khai thác đảm bảo tính toàn vẹn và bảo mật của dữ liệu. Điều này giúp ngăn chặn các cuộc tấn công và gian lận trong quá trình giao dịch, đồng thời tăng cường sự tin cậy và đáng tin cậy của hệ thống tiền điện tử.

Luận văn này tập trung vào nghiên cứu về cơ chế Mining và xây dựng một hệ thống Mining Asic trên FPGA (Field-Programmable Gate Array). Mining là quá trình tính toán và xác nhận các giao dịch trong mạng lưới tiền điện tử, đặc biệt là tiền điện tử sử dụng thuật toán băm (hashing) như Bitcoin, quá trình này giúp duy trì tính toàn vẹn của giao dịch và bảo vệ mạng lưới khỏi các cuộc tấn công.

Trong luận văn, ta nghiên cứu các thuật toán băm phổ biến như SHA-256 và xây dựng hệ thống Mining Asic trên FPGA để tối ưu hóa quá trình tính toán. Việc sử dụng FPGA cho phép tăng tốc độ tính toán và giảm thời gian cần thiết để tìm ra giải pháp cho bài toán mining.

Luận văn cũng đề cập đến các vấn đề liên quan đến tối ưu hóa và quản lý tài nguyên trong hệ thống Mining trên FPGA. Việc lập trình FPGA phải được thực hiện một cách cẩn thận để tận dụng tối đa khả năng tính toán của nó và đảm bảo rằng tài nguyên được sử dụng hiệu quả.

Tóm lại, luận văn này tập trung vào nghiên cứu cơ chế Mining và xây dựng hệ thống Mining Asic trên FPGA nhằm tăng tốc độ tính toán và hiệu suất trong quá trình Mining của tiền điện tử. Sự kết hợp giữa cơ chế Mining và công nghệ FPGA có thể cung cấp một giải pháp mạnh mẽ và hiệu quả cho việc khai thác tiền điện tử.

## MỤC LỤC

DANH MỤC CÁC HÌNH.....	1
DANH MỤC CÁC BẢNG.....	3
Chương 1 Giới thiệu.....	4
1.1. Giới thiệu đề tài.....	4
1.2. Nội dung nghiên cứu.....	4
1.3. Bố cục luận văn.....	5
Chương 2: Mining.....	6
2.1. Tổng quan về Mining.....	6
2.2. Khái niệm cơ bản trong Mining.....	8
2.2.1. P2P Network.....	8
2.2.2. Blockchain.....	9
2.2.3. Block Header.....	10
2.2.4 Target Threshold.....	12
2.2.5. Phương pháp Solo Mining.....	14
2.2.6. Phương pháp Pool Mining.....	16
2.2.7. Mining Software.....	17
2.2.8. Mining Hardware.....	19
Chương 3: Mô phỏng quá trình Mining trên Visual Studio.....	22
3.1. Thiết lập mô phỏng.....	22
3.2. Mining Software xử lý và gửi dữ liệu cho Mining Asic.....	23
3.3. Mining Asic xử lý gửi dữ liệu cho Mining Software.....	24

3.4.Kết quả mô phỏng .....	24
Chương 4: SHA256.....	27
4.1.Giới thiệu.....	27
4.2.Tổng quan về thuật toán SHA256 .....	27
4.2.1.Một số kí hiệu dùng trong thuật toán .....	27
4.2.2.Dán thêm dữ liệu(padding) .....	28
4.2.3.Mô hình thuật toán SHA256 .....	29
4.3. Chi tiết thuật toán SHA256 .....	29
4.3.1.Quá trình tiền xử lí .....	29
4.3.2.Tính toán băm SHA256 .....	31
4.4.Ứng dụng thuật toán SHA256 trong Mining mã hóa Block Header.....	33
4.5.IP core SHA256 .....	34
4.5.1.Đặc tả thiết kế .....	34
4.5.2.Sơ đồ khối SHA256_Core .....	35
4.5.3.Sơ đồ khối SHA256 .....	37
4.6.Testbench .....	39
4.6.1.Testbench cho một thông điệp .....	39
4.6.2.Testbench mô phỏng quá trình Mining .....	40
4.7.Đánh giá IP core SHA256.....	41
Chương 5: Xây dựng hệ thống Mining Asic trên FPGA .....	43
5.1.Tổng quan về DE10 Stander FPGA .....	43
5.2.Thiết kế phần cứng trên Platform Designer .....	44

5.3. Xây dựng phần mềm .....	46
5.3.1. Phân chia bộ nhớ cho mỗi khối tính toán SHA256.....	46
5.3.2. Xây dựng phần mềm .....	48
Chương 6: Thử nghiệm và đánh giá.....	50
6.1. Mục tiêu thử nghiệm .....	51
6.2. Thiết lập thử nghiệm .....	51
6.3. Kết quả thử nghiệm .....	52
6.4. Kết luận và hướng phát triển.....	57
GIẢI THÍCH THUẬT NGỮ.....	58
TÀI LIỆU THAM KHẢO.....	59



## DANH MỤC CÁC HÌNH

Hình 2.1 : Tổng quan về Mining.....	6
Hình 2.2 : P2P Network .....	8
Hình 2.3: Blockchain .....	10
Hình 2.4: Tổng quan Block trong Blockchain .....	10
Hình 2.5: nBits trong Block Header.....	13
Hình 2.6: Workflow của quá trình Solo Mining trong Bitcoin.....	14
Hình 2.7: Workflow của quá trình Pool Mining trong Bitcoin.....	16
Hình 2.8: Hoạt động của Mining Hardware.....	20
Hình 3.1: Kết quả xử lý từ Mining Software trên Visual Studio .....	25
Hình 3.2: Kết quả quá trình thử và sai của Mining Asic trên Visual Studio .....	26
Hình 3.3: Kết quả quá trình Mining trên Visual Studio .....	26
Hình 4.1: Mô hình thuật toán SHA256 .....	29
Hình 4.2: Khối tính toán băm SHA256.....	31
Hình 4.3: Quá trình mã hóa Block Header.....	33
Hình 4.4: Tổng quan thuật toán SHA256 .....	34
Hình 4.5: Top – view của khối SHA256_Core.....	35
Hình 4.6: Top-view của khối SHA256 .....	37
Hình 4.7: Kết quả testbench thông điệp chỉ 1 khối.....	39
Hình 4.8: Kết quả testbench thông điệp nhiều khối.....	40
Hình 4.9: Kết quả testbench quá trình Mining.....	41
Hình 4.10: Kết quả testbench quá trình Mining (wareform).....	41

Hình 4.11: Tốc độ băm của SHA256 .....	41
Hình 5.1: Tổng quan hệ thống.....	44
Hình 5.2: Chi tiết hệ thống trên Platform Designer .....	45
Hình 5.3: Tổng hợp tài nguyên đã sử dụng.....	46
Hình 5.4: Quá trình mã hóa Block Header.....	46
Hình 5.5: Phân chia bộ nhớ cho các core SHA256 trong bộ nhớ .....	47
Hình 5.6: Flowchart quá trình mining.....	49
Hình 6.1: Kết quả mining cho Block Header 1 .....	52
Hình 6.2: Kết quả mining cho Block Header 2.....	52
Hình 6.3: Kết quả mining cho Block Header 3 .....	53
Hình 6.4: Block 1 trên mạng blockchain (bitcoin).....	54
Hình 6.5: Block 2 trên mạng Blockchain (bitcoin).....	54
Hình 6.6: Block 3 trên mạng Blockchain (bitcoin).....	55
Hình 6.7: Khảo sát core SHA256 trên Signal Tap.....	55

## **DANH MỤC CÁC BẢNG**

Bảng 2.1: Các thành phần trong Block header.....	11
Bảng 4.1: Các tín hiệu của khối SHA256_Core .....	36
Bảng 4.2: Các tín hiệu của khối SHA256 .....	38
Bảng 6.1: Khảo sát tốc độ của hệ thống Mining .....	56

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. Giới thiệu đề tài:

Trong thời đại kỹ thuật số ngày càng phát triển, tiền điện tử đã thu hút sự chú ý của cả nhà đầu tư và công nghiệp tài chính. Điều đáng chú ý là quá trình khai thác tiền điện tử, tức quá trình tính toán và xác nhận giao dịch trong mạng lưới tiền điện tử, đóng vai trò quan trọng trong việc duy trì tính toàn vẹn và bảo mật của hệ thống.

Hiện nay, các thiết bị đào ASIC (Application-Specific Integrated Circuit) đang chiếm ưu thế trong hoạt động khai thác tiền điện tử. ASIC là các chip tích hợp được thiết kế đặc biệt để thực hiện các nhiệm vụ khai thác tiền điện tử một cách hiệu quả và tối ưu.

Tuy nhiên, một điểm đáng lưu ý là chi phí của các thiết bị đào ASIC hiện nay rất đắt đỏ. Chi phí cao của các thiết bị đào ASIC bao gồm giá mua ban đầu của thiết bị và chi phí điện năng liên quan đến việc vận hành chúng. Các thiết bị đào ASIC có khả năng tính toán cao hơn so với các phương pháp khai thác truyền thống, nhưng đồng thời cũng tiêu thụ lượng điện năng lớn. Điều này tạo ra một chi phí năng lượng đáng kể trong việc vận hành thiết bị đào và có thể ảnh hưởng đến lợi nhuận cuối cùng của người khai thác.

Công nghệ FPGA linh hoạt và có thể được lập trình lại, mở ra khả năng tối ưu hóa quá trình khai thác với chi phí tài chính và năng lượng hợp lý hơn. Điều này có thể mang lại sự cân bằng và đa dạng hóa trong hoạt động khai thác tiền điện tử.

## 1.2. Nội dung nghiên cứu

Luận văn cần tìm hiểu: các hình thức đào tiền điện tử, tìm hiểu về cơ chế Mining sử dụng thuật toán SHA256, xây dựng lõi IP SHA256, từ đó xây dựng một hệ thống Mining Asic dựa trên nền tảng FPGA

### **1.3.   Bố cục luận văn**

- Chương 1: Giới thiệu đề tài
- Chương 2: Trình bày về cơ chế Mining và các khái niệm liên quan đến Mining tiền điện tử
- Chương 3: Thực hiện mô phỏng quá trình Mining trên Visual Studio
- Chương 4: Tìm hiểu về thuật toán mã hóa SHA256 được sử dụng trong Mining tiền điện tử, xây dựng lõi IP SHA256
- Chương 5: Xây dựng hệ thống Mining Asic dựa trên nền tảng FPGA
- Chương 6: Trình bày thử nghiệm, đánh giá kết quả, kết luận và hướng phát triển của luận văn

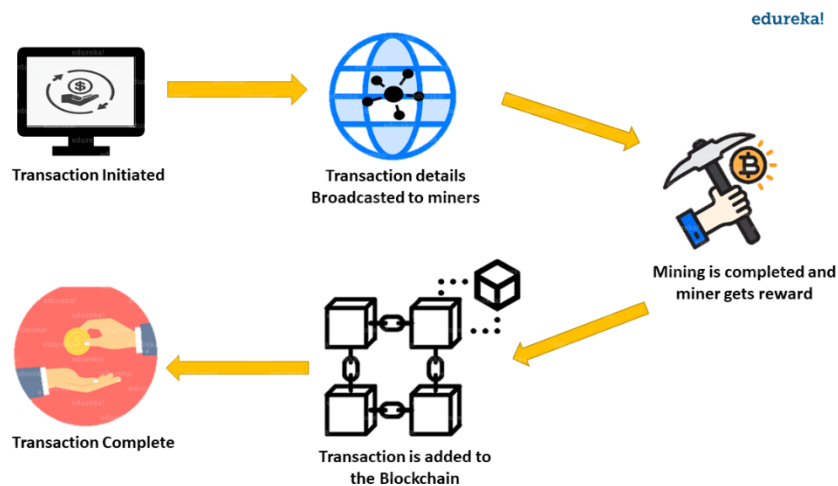
## CHƯƠNG 2: MINING

### 2.1. Tổng quan về Mining

Mining (khai thác) là quá trình xác nhận và xử lý các giao dịch trong mạng lưới tiền điện tử. Các máy tính hoặc thiết bị đặc biệt (như ASIC hoặc FPGA) sẽ tham gia vào việc giải quyết các bài toán tính toán phức tạp để xác nhận các giao dịch. Bài toán này thường liên quan đến việc tìm ra một giá trị hash (mã băm) duy nhất thỏa mãn một số yêu cầu cụ thể.

Người tham gia vào quá trình mining (miner) sẽ cạnh tranh với nhau để tìm ra giá trị hash đầu tiên và chính xác nhất. Người thành công sẽ được thưởng bằng một số lượng đồng tiền điện tử mới và có thể nhận được các khoản phí giao dịch.

Quá trình mining không chỉ xác nhận giao dịch mà còn giúp bảo vệ mạng lưới tiền điện tử khỏi các cuộc tấn công và gian lận



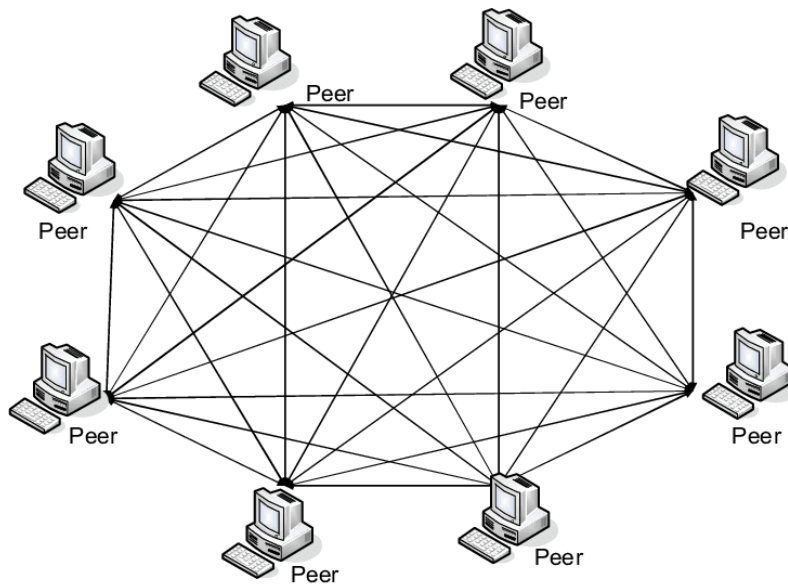
Hình 2.1 : Tổng quan về Mining

- Hiện nay, có 2 phương pháp Mining chính:
  - + Solo mining: là phương pháp mà một cá nhân hoặc một nhóm nhỏ tự mình khai thác tiền điện tử. Người khai thác giải quyết các bài toán tính toán phức tạp và xác nhận các giao dịch trực tiếp trên mạng lưới tiền điện tử. Phần thưởng được nhận nếu thành công là toàn bộ cho người khai thác.
  - + Pool mining: Đây là phương pháp mà nhiều người khai thác hợp tác với nhau trong một nhóm (pool). Cùng giải quyết các bài toán tính toán, khi một giá trị hash đúng được tìm thấy, phần thưởng sẽ được chia đều giữa các thành viên theo tỷ lệ đóng góp tính toán của mỗi người

## 2.2. Khái niệm cơ bản trong Mining

### 2.2.1. P2P Network

P2P network (Peer-to-Peer network) là một hệ thống mạng được sử dụng trong blockchain, đây là hệ thống mạng phân tán, trong đó các nút (nodes) trong blockchain kết nối và giao tiếp trực tiếp với nhau để truyền tải thông tin và dữ liệu liên quan đến giao dịch và khối (block).



Hình 2.2 : P2P Network

Trong mạng P2P của blockchain, mỗi node trong hệ thống hoạt động như một đối tác bình đẳng, không có sự phân cấp giữa các node. Các node trong mạng gửi và nhận thông tin với nhau thông qua giao thức mạng, chia sẻ dữ liệu về giao dịch mới, khối mới và các thông tin liên quan khác.

Qua mạng P2P, các node trong blockchain có thể xác minh và đồng bộ hóa thông tin với nhau, xác nhận tính hợp lệ của giao dịch và xây dựng chuỗi khối chung. Khi một giao dịch mới được tạo ra, nó được phát tán qua mạng P2P đến các nút khác để được xác nhận



và đưa vào khối mới. Các nút trong mạng đồng thời nhận, xác minh và lưu trữ các khối mới trong blockchain.

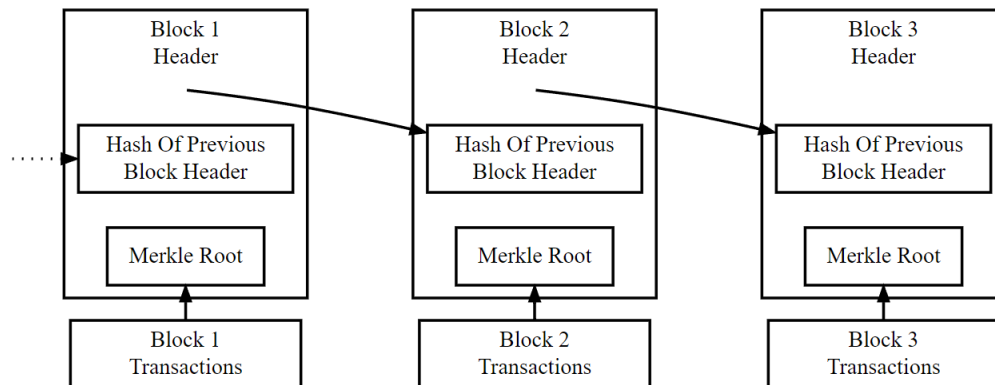
Ưu điểm của việc sử dụng P2P Network trong blockchain:

- Phân tán và không có điểm trung tâm : không có một thực thể duy nhất kiểm soát, các nút trong mạng có vai trò bình đẳng và đóng góp vào việc duy trì hệ thống.
- Khả năng chống tấn công: nhiều nút kiểm tra và xác minh thông tin, việc thực hiện một cuộc tấn công thành công đòi hỏi sự chiếm đoạt lớn về khả năng tính toán của mạng, điều này rất khó và tốn kém.
- Tính minh bạch và kiểm tra công khai: mỗi node trong mạng có thể xem và kiểm tra thông tin từ các node khác, đảm bảo tính toàn vẹn và minh bạch của dữ liệu.

### **2.2.2. Blockchain**

Blockchain là một công nghệ đặc biệt, phục vụ như một “cuốn sổ cái” công khai. Đó là một cơ sở dữ liệu phi tập trung và phân tán chứa một bản ghi đã được sắp xếp và ghi dấu thời gian về tất cả các giao dịch được thực hiện trong mạng lưới.

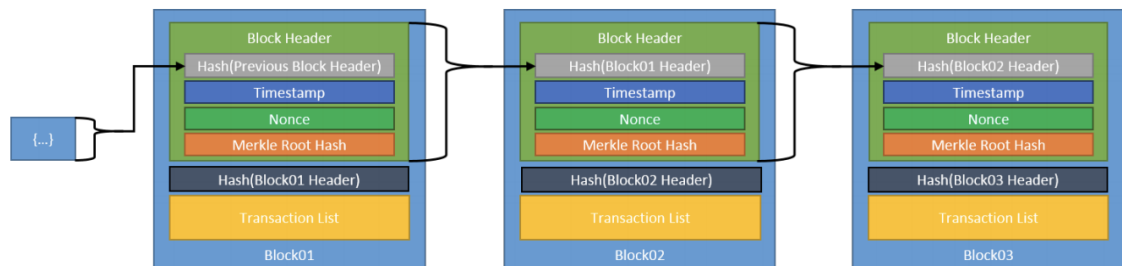
Cấu trúc cơ bản của blockchain là một chuỗi các khối (block), trong đó mỗi block chứa một tập hợp các giao dịch và thông tin khác. Mỗi khối được liên kết với khối trước đó thông qua một mã băm (hash), tạo thành một chuỗi liên kết không thể thay đổi.



Hình 2.3: Blockchain

Blockchain được xây dựng trên nguyên tắc của Proof of Work (POW). Khi một node trong mạng lưới muốn tạo một block mới. Node này sẽ bắt đầu tính toán một giá trị băm (hash) cho block đó. Mục tiêu là tìm ra một giá trị băm sao cho nó nhỏ hơn một ngưỡng (threshold) nhất định. Khi node tìm được đáp án đúng, nó sẽ công bố cho các node khác trong mạng lưới, các node sẽ thực hiện xác nhận tính đúng đắn của block và cuối cùng là block được thêm vào blockchain.

### 2.2.3 Block Header



Hình 2.4 : Tổng quan block trong blockchain

Một block chứa các thông tin cơ bản sau:

- Block Header: một phần quan trọng của block, chứa thông tin để xác định và

- Body: chứa danh sách các giao dịch (transactions) và các thông tin khác liên quan đến hoạt động của blockchain. Các giao dịch được thêm vào block bao gồm thông tin về người gửi, người nhận, số lượng tiền và các thông tin khác liên quan.
- Hash: một giá trị hash duy nhất được tạo bằng cách áp dụng một thuật toán hash lên dữ liệu của block (bao gồm cả block header và body). Giá trị hash này định danh và xác thực tính toàn vẹn của block.
- Liên kết giữa các block: mỗi block trong blockchain được liên kết với block trước đó thông qua giá trị hash của block trước đó trong block header. Điều này tạo ra một chuỗi các block liên tiếp và đảm bảo tính bất biến của blockchain.

Block header là một phần quan trọng của mỗi block trong blockchain. Nó chứa các thông tin cơ bản và khối lượng dữ liệu nhỏ hơn so với toàn bộ khối. Block header giúp định danh và xác định tính hợp lệ của khối đó.

Các thành phần chính của block header:

Bảng 2.1: Các thành phần trong Block header

Tên	Ý nghĩa	Độ dài (byte)
Version(Phiên bản)	Số phiên bản khối	4
Previous Hash(Địa chỉ khối trước)	Giá trị hash của block trước đó trong blockchain, liên kết các block với nhau và tạo thành một chuỗi blockchain.	32
Merkle Root(Gốc mekle)	Giá trị hash của cây Merkle, được sử dụng để đại diện cho tất cả các giao dịch trong khối	32

Timestamp	Thời gian khối được tạo ra	4
nBits	Giá trị mục tiêu về độ khó của quá trình Proof of Work. Nó xác định mức độ khó của việc tìm một giá trị hash hợp lệ cho khối.	4
Nonce	Số được thay đổi lặp đi lặp lại trong quá trình khai thác khối để tạo ra một giá trị hash thỏa mãn yêu cầu độ khó của mạng	4

Thông qua việc kết hợp các thành phần trên, block header tạo nên một chuỗi số liệu được hash thành một giá trị duy nhất, được gọi là "Block Header Hash". Quá trình Mining đòi hỏi các máy tính trong mạng lưới phải thử nghiệm nhiều giá trị nonce khác nhau để tìm ra một Block Header Hash thỏa mãn yêu cầu độ khó. Khi một máy tính khai thác thành công, nó gửi block header và nonce đã tìm thấy cho các node khác trong mạng lưới để được xác nhận và thêm vào blockchain.

#### **2.2.4. Target Threshold**

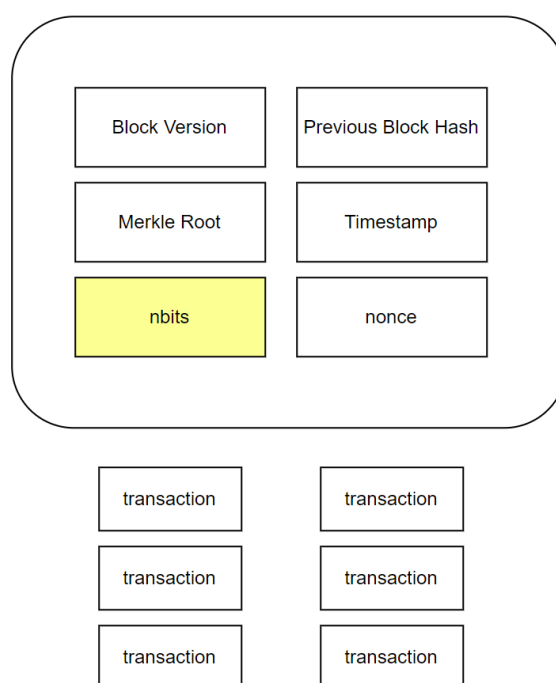
Target Threshold là giá trị số được sử dụng để kiểm tra tính hợp lệ của một khối mới được khai thác.

Trong quá trình khai thác, các miner sẽ phải tìm ra một số Nonce sao cho giá trị băm (hash) của block là một số nhỏ hơn hoặc bằng giá trị Target Threshold. Target Threshold được biểu thị dưới dạng một số nguyên không dấu 256-bit.

Mạng Bitcoin sẽ điều chỉnh giá trị Target Threshold dựa trên tỷ lệ thời gian trung bình giữa các block mới được thêm vào. Mục tiêu là duy trì thời gian trung bình giữa các

block một khoảng thời gian nhất định. Chẳng hạn với mạng Bitcoin là 10 phút. Nếu quá trình mining khối trước đó mất nhiều thời gian hơn 10 phút, giá trị Target Threshold sẽ được điều chỉnh giảm để làm cho quá mining dễ dàng hơn. Ngược lại, nếu quá trình mining mất ít thời gian hơn 10 phút, giá trị Target sẽ được điều chỉnh tăng để làm cho quá trình mining trở nên khó hơn.

nBits là một trường trong block để đại diện cho Target Threshold trong quá trình mining, đây là trường 32-bit, chuyển đổi từ nBits sang Target Threshold là để xác định độ khó của việc đào một block trong mạng Blockchain.



Hình 2.5: nbits trong Block header

Target Threshold được tính theo định dạng “compact”. Nghĩa là trong 4 byte của nBits thì 1 byte đầu tiên là bước mũ (exponent) và 3 byte còn lại đại diện cho hệ số (coefficient). Công thức để tính toán Target Threshold:

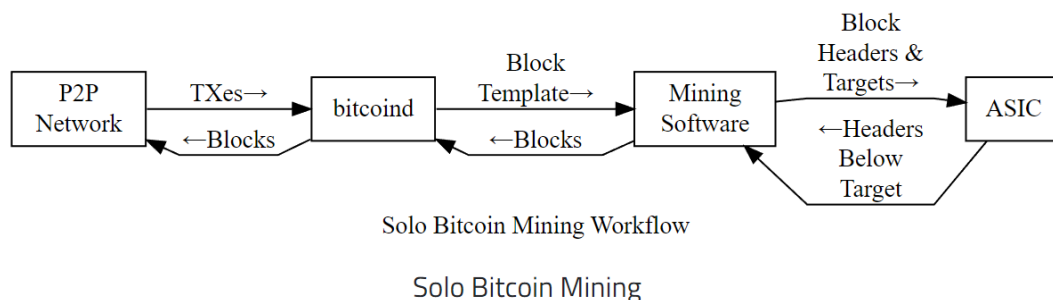
$$\text{Target} = \text{coefficient} * 256 ^ ( \text{exponent} - 3)$$

Ví dụ: với nBits = 0x18bc330

[illegible]

### 2.2.5. Phương pháp Solo Mining

Solo mining là quá trình khi một người tham gia vào mạng blockchain và khai thác các block mới một cách độc lập. Trong Solo mining, các Miner làm việc một mình để tìm ra giải pháp cho một bài toán hash phức tạp nhằm xác nhận các giao dịch và thêm một block mới vào Blockchain. Hình 2.6 là Flow biểu diễn quá trình Mining của mạng Bitcoin.



Hình 2.6: Workflow của quá trình Solo Mining trong Bitcoin

Quá trình Solo Mining trong Bitcoin gồm 4 bước chính:

1. Các thợ đào sử dụng bitcoind<sup>(1)</sup> để lấy các giao dịch mới từ mạng Bitcoin
  2. Mining software tạo khối mới từ khối mẫu
  3. Mining software xây dựng block header và gửi đến cho ASIC
  4. Mining hardware thực hiện quá trình thử và sai (trial and error)
- a. Các thợ đào sử dụng bitcoind để lấy các giao dịch mới từ mạng Bitcoin**

- Các thợ đào sử dụng bitcoind để tải xuống và đồng bộ hóa blockchain của Bitcoin từ mạng Bitcoin
- Các giao dịch mới được phát hiện trên mạng Bitcoin thông qua các node khác nhau.
- Bitcoind cập nhật danh sách các giao dịch chưa được xác nhận trong mempool<sup>(2)</sup>, cung cấp cho các thợ đào thông tin để tạo các khối mới trong quá trình đào Bitcoin

***b. Mining software tạo khối mới từ khối mẫu:***

- Mining software có thể lấy được các thông tin sau từ khối mẫu để tạo khối mới:
  - + Các giao dịch trong mempool của bitcoind
  - + Các thông tin cần thiết để tạo một coinbase transaction trả tiền thưởng cho người đào
  - + Các thông tin cần thiết để tạo tiêu đề khối cho khối tiếp theo, bao gồm block version, previous block hash, and bits (target).

***c. Mining software xây dựng block header và gửi đến cho ASIC***

- Mining software sử dụng các thông tin được thu thập được ở bước trên để xây dựng Block header.
- Block header được mã hóa dưới dạng chuỗi các byte có kích thước 80 byte gồm các thành phần: version, previous block header hash, merkle root hash, time, bBits, nonce.
- Sau đó, Mining software gửi 80-byte block header đến ASIC ( mining hardware) cùng với target threshold ( ngưỡng)

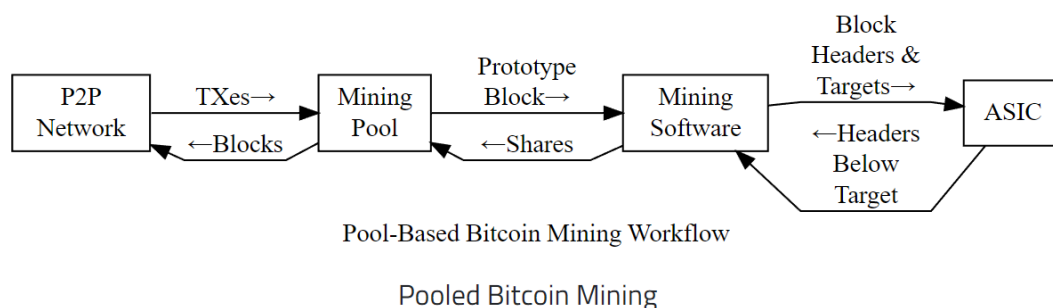
***d. Mining hardware thực hiện quá trình trial and error ( thử và sai):***

- Mining hardware lặp lại mọi giá trị có thể có của số Nonce của Block header và tạo mã hash tương ứng:

- + Nếu một hash được tìm thấy dưới target threshold, mining hardware sẽ trả lại block header có số nonce thành công cho software mining. Software mining sẽ kết hợp header với khối và gửi khối hoàn thành đến bitcoind để được thêm vào blockchain.
- + Nếu không có bất kỳ hash nào thấp hơn target threshold, mining hardware sẽ nhận được một block header mới với merkle root mới từ software mining. Block header mới này được tạo ra bằng cách thêm dữ liệu extra nonce vào field coinbase của coinbase transaction.

### 2.2.6. Phương pháp Pool Mining

Pool Mining là phương pháp mining mà trong đó một nhóm các miner kết hợp sức mạnh tính toán với nhau để tăng khả năng tìm khối mới và chia sẻ phần thưởng. Thay vì đào một mình và cạnh tranh với các miner khác, Pool Mining cho phép các miner làm việc nhóm để tăng hiệu suất đào và cải thiện khả năng nhận phần thưởng.



Hình 2.7: Workflow của quá trình Pool Mining trong Bitcoin

Quá trình đào Bitcoin trong pool tương tự như solo mining, nhưng có một số khác biệt:

- Mining Pool<sup>(3)</sup> sử dụng bitcoind để lấy giao dịch mới nhất từ mạng Bitcoin.
- Mining software của các miner kết nối với pool và yêu cầu các thông tin để tạo block headers.



- Trong pool mining, ngưỡng (Target Threshold) được đặt thấp hơn nhiều lần so với độ khó của mạng, điều này giúp cho Mining Hardware trả về nhiều Block Headers mà không phải làm việc đến mức được thêm vào blockchain. Tuy nhiên, các block headers này lại đủ điều kiện để được thêm vào Pool Mining.
- Thông tin mà thợ đào gửi đến pool được gọi là "shares", điều này sẽ chứng minh rằng miner đó đã thực hiện một phần công việc của pool. Các share này được sử dụng để tính toán số lượng tiền thưởng được trả cho miner trong pool.
- Nếu pool mining tìm ra một block đủ điều kiện để được thêm vào blockchain của mạng, nó sẽ gửi block này đến mạng để được thêm vào blockchain và nhận phần thưởng và phí giao dịch tương ứng.

### ***2.2.7. Mining Software***

Mining Software là một phần mềm được sử dụng để thực hiện mining bằng cách sử dụng sức mạnh tính toán của các thiết bị đào

- Chức năng: Mining software được thiết kế để thực hiện các nhiệm vụ quan trọng trong quá trình đào Bitcoin, bao gồm:
  - + Giao tiếp với Mining Pool hoặc mạng ngang hàng: Mining software cần kết nối và giao tiếp với Mining Pool (nếu sử dụng pool mining) hoặc mạng ngang hàng Bitcoin để nhận thông tin về block mới, giao dịch và gửi kết quả đào.
  - + Xác minh tính hợp lệ của giao dịch và khối: Mining software sẽ kiểm tra tính hợp lệ của giao dịch và khối thông qua việc xác minh chữ ký số, địa chỉ ví và các quy tắc khác của mạng.
  - + Ghi nhận và báo cáo kết quả đào: Khi mining hardware tìm ra một nonce hợp lệ và tạo ra mã băm thỏa mãn yêu cầu, nó sẽ ghi nhận kết quả này và báo cáo lại cho Mining Pool (nếu sử dụng pool mining) hoặc lưu trữ kết quả trên máy tính cá nhân.

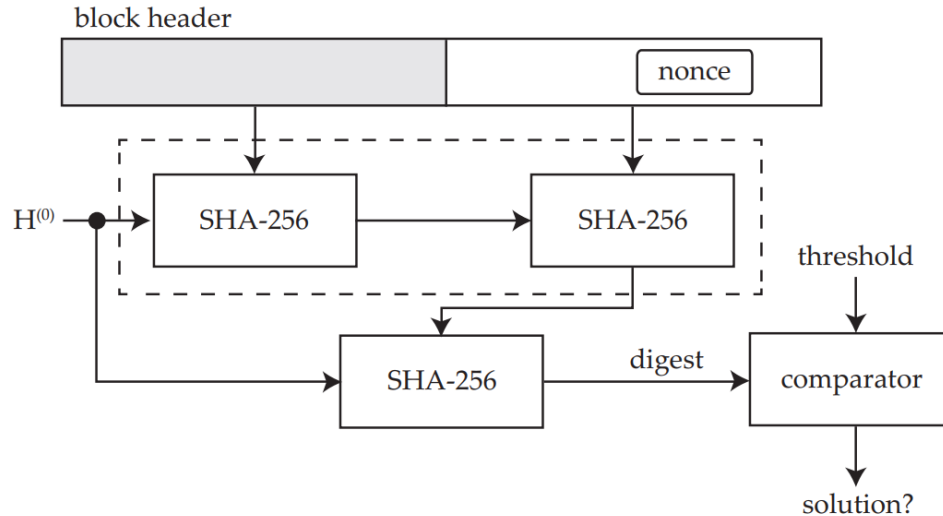
- Đa nền tảng: Mining software có sẵn trên nhiều nền tảng khác nhau, bao gồm hệ điều hành Windows, macOS và Linux. Điều này cho phép người dùng sử dụng Mining Software trên các máy tính cá nhân, máy chủ hoặc thiết bị đào chuyên dụng.
- Hiệu suất và tùy chỉnh: Mining software cung cấp các tùy chọn tùy chỉnh để người dùng có thể điều chỉnh hiệu suất và cấu hình của quá trình đào. Người dùng có thể điều chỉnh công suất tính toán, tần số, tối ưu hóa các tham số và quản lý các thiết bị đào.
- Các Mining Software phổ biến hiện nay:
  - + CGMiner: là một phần mềm đào mã nguồn mở phát triển bởi Con Kolivas. Đây là một trong những phần mềm đào phổ biến nhất trong cộng đồng đào Bitcoin và đã tồn tại từ năm 2011. Hỗ trợ nhiều hệ điều hành và các thiết bị đào khác nhau như Asic, FPGA.
  - + BFGMiner: phần mềm đào mã nguồn mở, phát triển bởi Luke-Jr và được công bố lần đầu vào năm 2011. BFGMiner là một trong những phần mềm đào phổ biến và được sử dụng rộng rãi trong cộng đồng.
  - + EasyMiner: Một phần mềm đào dựa trên giao diện đồ họa (GUI) dễ sử dụng, hỗ trợ cho người mới bắt đầu trong việc đào tiền điện tử.
  - + BitMinter: Một phần mềm đào đám mây và Pool Mining đơn giản và dễ sử dụng.

### ***2.2.8. Mining Hardware/***

Mining hardware là thành phần quan trọng trong quá trình mining, nó đóng vai trò quyết định đến hiệu suất và lợi nhuận của việc đào. Mining Hardware là các thiết bị được sử dụng để tính toán và giải mã các khối giao dịch, tạo ra các khối mới trong chuỗi khối của mạng tiền điện tử.

Vai trò chính của Mining Hardware bao gồm:

- + Nhận thông tin: Mining hardware nhận thông tin về Block Header và Target Threshold từ Mining Software. Block Header chứa các thông tin về khối giao dịch và các dữ liệu khác cần thiết để tạo ra Block mới.
- + Thực hiện quá trình thử và sai: Mining Hardware sử dụng thuật toán mã hóa SHA-256 (Secure Hash Algorithm 256-bit) để mã hóa Block Header với số nonce trong khoảng giá trị 4 byte từ 0x00000000 đến 0xFFFFFFFF. Bằng cách thay đổi giá trị nonce trong khoảng đó và tiến hành hash, Mining Hardware tạo ra các mã hash khác nhau.
- + So sánh với Target Threshold: Mỗi lần Mining Hardware tạo ra một mã hash, nó so sánh mã hash này với Target Threshold đã nhận được từ Mining Software. Nếu mã hash nhỏ hơn hoặc bằng Target, Block Header đó được coi là hợp lệ.
- + Gửi kết quả về Mining Software: Khi Mining Hardware tìm ra một Block Header hợp lệ, nó gửi kết quả này về cho Mining Software để được xác nhận và đưa vào Blockchain của mạng tiền điện tử.



Hình 2.8: Hoạt động của Mining Hardware

Các loại Mining Hardware:

- a) CPU (Central Processing Unit): CPU là phần cứng phổ biến nhất và đã được sử dụng trong giai đoạn đầu của việc đào tiền điện tử. Tuy nhiên, với sự phát triển của mạng tiền điện tử và tăng đáng kể về độ khó của quá trình đào, CPU trở nên không đủ mạnh mẽ để đáp ứng yêu cầu hiện tại.
- b) GPU (Graphics Processing Unit): GPU đã thay thế CPU trong việc đào tiền điện tử. GPU có khả năng xử lý song song cao hơn, với số lượng lõi tính toán lớn hơn và hiệu suất tốt hơn so với CPU. Điều này cho phép GPU đào được nhiều đồng tiền điện với hiệu suất tốt hơn.
- c) ASIC (Application-Specific Integrated Circuit): ASIC là các chip đặc chủng được thiết kế đặc biệt cho mục đích đào tiền điện tử. ASIC có khả năng tính toán mạnh mẽ và hiệu suất cao hơn rất nhiều so với CPU và GPU. Nhờ tính chất đặc thù của nó, ASIC được tối ưu hóa để thực hiện các phép tính hash liên quan đến quá trình đào tiền điện tử. Các thiết bị ASIC hiện nay: SP20 Jackson, Antminer R4, Antminer S5, Antminer S9,...

d) FPGA (Field-Programmable Gate Array) là một loại phần cứng có khả năng lập trình lại. Nó được sử dụng để tối ưu hiệu suất tính toán và tiêu thụ năng lượng thấp trong quá trình đào tiền điện tử. FPGA đa năng và linh hoạt, cho phép người dùng tùy chỉnh và tối ưu hóa cho các thuật toán đào cụ thể, từ đó có thể tối ưu hóa tốc độ đào.

## Chương 3: Mô phỏng quá trình Mining trên Visual Studio

### 3.1. Thiết lập mô phỏng

Chúng ta sẽ thực hiện thiết lập cho 2 quá trình chính là:

- + Quá trình Mining Software nhận, xử lý dữ liệu và gửi dữ liệu cho Mining ASIC
- + Quá trình Mining ASIC nhận và thực hiện Mining

#### *a. Thiết lập thứ nhất:*

Mining Software sẽ nhận Block Header từ Mining Pool ( với phương pháp Pool Mining ) hoặc từ bitcoind ( với Solo Mining ). Trong mô phỏng này, chúng ta thiết lập Block Header mà Mining Software nhận được ở trong file BlockHeader.txt

Giá trị Block Header (80 byte) thực hiện mô phỏng là:

```
0x010000009810f0fa1817a4d2d371a069addaafab2ca99887abcc5bd2528e4341000000  
00654f005a6e4b4b57b42343fb0e47f32079b4ebfe643c2ea4ea20e46c3af00c238d46684  
9ffff001d000000000
```

Mining Software sau khi nhận được Block Header sẽ thực hiện tính toán Target Threshold và gửi Block Header cùng với Target Threshold cho Mining ASIC. Trong mô phỏng này, Block Header và giá trị Target Threshold tính toán được sẽ được lưu vào file DataSendFPGA.txt

#### *b. Thiết lập thứ hai:*

Mining ASIC sẽ nhận Block Header và Target Threshold từ Mining Software gửi đến. Trong mô phỏng này, dữ liệu mà Mining ASIC nhận được sẽ là dữ liệu trong file DataSendFPGA.txt – đây chính là dữ liệu mà Mining Software tính toán để gửi cho Mining ASIC.

Mining ASIC sẽ thực hiện quá trình thử và sai để tìm ra block header hợp lệ.

### 3.2. Mining Software xử lý và gửi dữ liệu cho Mining ASIC

Block Header mà Mining Software nhận được là:

0x010000009810f0fa1817a4d2d371a069addaafab2ca99887abcc5bd2528e4341000000  
00654f005a6e4b4b57b42343fb0e47f32079b4ebfe643c2ea4ea20e46c3af00c238d46684  
9ffff001d00000000

Block Header trên sẽ gồm các thành phần sau:

Tên	Độ dài (byte)	Giá trị ( little endian )
Version(Phiên bản)	4	0x01000000
Previous Hash(Địa chỉ khối trước)	32	9810f0fa1817a4d2d371a069addaafab2ca99887abcc5bd2528e434100000000
Merkle Root(Gốc mekle)	32	654f005a6e4b4b57b42343fb0e47f32079b4ebfe643c2ea4ea20e46c3af00c23
Timestamp	4	8d466849
nBits	4	ffff001d
Nonce	4	00000000







```
D:\My_Data\KLTN\visual stud  X + -
Nounce = 48ac is invalid! Hashrate = 4651
Nounce = 48ad is invalid! Hashrate = 4651.25
Nounce = 48ae is invalid! Hashrate = 4651.5
Nounce = 48af is invalid! Hashrate = 4651.75
Nounce = 48b0 is invalid! Hashrate = 4652
Nounce = 48b1 is invalid! Hashrate = 4652.25
Nounce = 48b2 is invalid! Hashrate = 4652.5
Nounce = 48b3 is invalid! Hashrate = 4652.75
Nounce = 48b4 is invalid! Hashrate = 4653
Nounce = 48b5 is invalid! Hashrate = 4653.25
Nounce = 48b6 is invalid! Hashrate = 4653.5
Nounce = 48b7 is invalid! Hashrate = 4653.75
Nounce = 48b8 is invalid! Hashrate = 4654
Nounce = 48b9 is invalid! Hashrate = 4654.25
Nounce = 48ba is invalid! Hashrate = 4654.5
Nounce = 48bb is invalid! Hashrate = 4654.75
Nounce = 48bc is invalid! Hashrate = 4655
Nounce = 48bd is invalid! Hashrate = 4655.25
Nounce = 48be is invalid! Hashrate = 4655.5
Nounce = 48bf is invalid! Hashrate = 4655.75
Nounce = 48c0 is invalid! Hashrate = 4656
Nounce = 48c1 is invalid! Hashrate = 4656.25
Nounce = 48c2 is invalid! Hashrate = 4656.5
Nounce = 48c3 is invalid! Hashrate = 4656.75
Nounce = 48c4 is invalid! Hashrate = 4657
Nounce = 48c5 is invalid! Hashrate = 4657.25
Nounce = 48c6 is invalid! Hashrate = 4657.5
Nounce = 48c7 is invalid! Hashrate = 4657.75
```

Hình 3.2: Kết quả quá trình thử và sai của Mining Asic trên Visual Studio

```
Microsoft Visual Studio Debu  X + -
Nounce = b38c4c21 is invalid! Hashrate = inf
Nounce = b38c4c22 is invalid! Hashrate = inf
Nounce = b38c4c23 is invalid! Hashrate = inf
Nounce = b38c4c24 is invalid! Hashrate = inf
Nounce = b38c4c25 is invalid! Hashrate = inf
Nounce = b38c4c26 is invalid! Hashrate = inf
Nounce = b38c4c27 is invalid! Hashrate = inf
Nounce = b38c4c28 is invalid! Hashrate = inf
Nounce = b38c4c29 is invalid! Hashrate = inf
Nounce = b38c4c2a is invalid! Hashrate = inf
Nounce = b38c4c2b is invalid! Hashrate = inf
Nounce = b38c4c2c is invalid! Hashrate = inf
Nounce = b38c4c2d is invalid! Hashrate = inf
Nounce = b38c4c2e is invalid! Hashrate = inf
Nounce = b38c4c2f is invalid! Hashrate = inf
Nounce = b38c4c30 is invalid! Hashrate = inf
Nounce = b38c4c31 is invalid! Hashrate = inf
Nounce = b38c4c32 is invalid! Hashrate = inf
Nounce = b38c4c33 is invalid! Hashrate = inf
Nounce = b38c4c34 is invalid! Hashrate = inf
Nounce = b38c4c35 is invalid! Hashrate = inf
SUCCESS!
Nounce = 3012316214 is valid! Hashrate = inf
Block Header below Target: 0000000071350772f98f84babf35502b33d42ee8466d3dde0f376c4120352081
D:\My_Data\KLTN\visual studio\CreateHashBlockHeader\x64\Release\CreateHashBlockHeader.exe (process 15296) exited with co
de 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Hình 3.3: Kết quả quá trình Mining trên Visual Studio

## Chương 4: SHA256

### 4.1. Giới thiệu

SHA256 là một thuật toán băm mạnh được sử dụng rộng rãi để tạo ra các giá trị băm duy nhất cho dữ liệu đầu vào. Nó là một phần của họ thuật toán băm SHA2 (bao gồm cả SHA-224, SHA-256, SHA-384, và SHA-512), được phát triển bởi Cục Tiêu Chuẩn và Công Nghệ Quốc Gia Hoa Kỳ (NIST) và công bố vào năm 2001.

Thuật toán băm bảo mật mã SHA2 đóng một vai trò quan trọng trong việc phát triển xã hội thông minh, nó cần thiết cho mục đích bảo mật dữ liệu và toàn vẹn dữ liệu trong nhiều ứng dụng, chẳng hạn như thuật toán chữ ký số (DSA), xác thực tin nhắn dựa trên băm (HMAC), tạo số giả ngẫu nhiên (PRNG) và xác thực thông báo trong bảo mật giao thức internet (IPSec). Ngoài ra, hàm băm SHA256 đã trở thành thành phần chính để bảo mật các mạng blockchain phi tập trung như là Bitcoin.

### 4.2. Tổng quan về thuật toán SHA256

#### 4.2.1. Một số kí hiệu dùng trong thuật toán

- $a, b, c, d, e, f, g, h$  Các biến dùng trong tính toán giá trị băm.
- $H^{(i)}$  Giá trị băm thứ  $i$ .  $H^{(0)}$  giá trị băm khởi tạo ban đầu,  $H^{(N)}$  giá trị băm cuối cùng.
- $H_j^{(i)}$  word thứ  $j$  của giá trị băm thứ  $i$
- $K_t$  hằng số được sử dụng trong tính toán giá trị băm
- $k$  số bit 0 được thêm vào trong quá trình padding
- $l$  chiều dài của thông điệp  $M$  (bit)
- $M$  Thông điệp cần được băm
- $n$  số bit được dịch (shift) hoặc xoay (rotated)
- $N$  số khối trong quá trình padding
- $W_t$  khối dữ liệu 32 bit đại diện cho 1 word

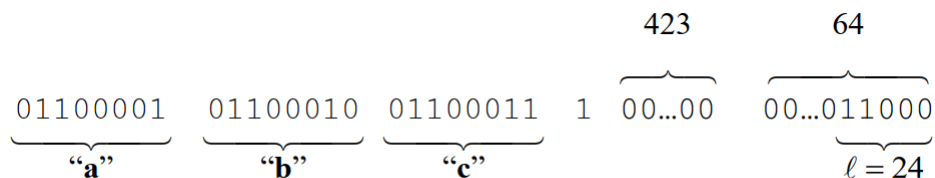
### Các toán tử sử dụng trong thuật toán:

- $\wedge$  AND bit
- $\vee$  OR bit
- $\oplus$  XOR
- $\neg$  đảo bit
- $+$  cộng modulo
- $\ll$  dịch bit sang trái
- $\gg$  dịch bit sang phải
- $ROTL^n(x)$  Rotate Left, phép xoay trái n bit trên một word x
- $ROTR^n(x)$  Rotate Right, phép xoay phải n bit trên một word x
- $SHR^n(x)$  Right Shift, dịch sang phải n bit trên một word x

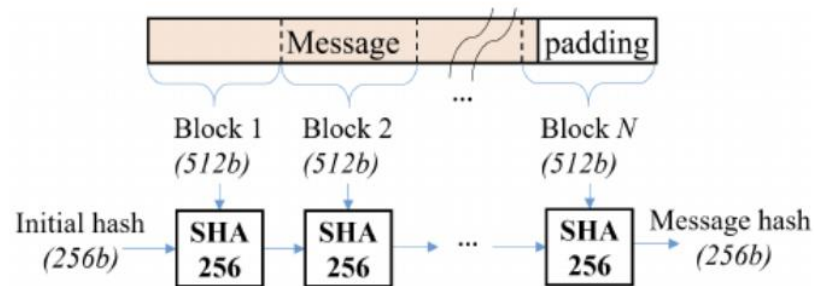
#### 4.2.2. Dán thêm dữ liệu (padding)

Padding của thông điệp  $M$  là chuỗi  $x$  bits được thêm vào sau thông điệp  $M$  để đạt được một bội số của kích thước khối. Để thực hiện được điều này cần phải biết độ dài chuỗi bit của  $M$  và kích thước một khối trong thuật toán.

Quy tắc padding như sau: Thêm một bit 1, tiếp theo là số bit 0 nhỏ nhất và 64 bit cuối cùng là độ dài của thông điệp sao cho thỏa điều kiện kích thước thông điệp là một bội số của kích thước khối.



### 4.2.3. Mô hình thuật toán SHA256



Hình 4.1: Mô hình thuật toán SHA256

Hình 4.1 là mô hình tổng thể cho thuật toán SHA256, thông điệp ban đầu sẽ được chia thành các khối dữ liệu 512 bit. Nếu khối cuối cùng nhỏ hơn 512 bit thì sẽ thực hiện dán dữ liệu (padding) thêm vào. Khối SHA256 sẽ tính toán các giá trị hàm băm trung gian cho từng khối dữ liệu 512 bit.

Kết quả hàm băm của khối hiện tại sẽ trở thành hàm băm ban đầu làm đầu vào để tính toán hàm băm của khối dữ liệu 512 bit tiếp theo. Kết quả của khối dữ liệu cuối cùng được coi là giá trị băm của toàn bộ thông điệp.

## 4.3. Chi tiết thuật toán SHA256

### 4.3.1. Quá trình tiền xử lý

Quá trình tiền xử lý gồm 3 bước:

- Dán dữ liệu vào thông điệp (padding)
- Chia thông điệp thành các khối
- Thiết lập giá trị băm khởi tạo  $H^{(0)}$

#### a. Dán dữ liệu vào thông điệp

Dữ liệu vào của thuật toán SHA256 là bội của 512 bit, mục đích của việc padding là thêm một số bit vào đằng sau thông điệp để đảm bảo thông điệp có độ dài là bội của 512bit.

Quy tắc padding: Thêm một bit 1, tiếp theo là số bit 0 nhỏ nhất và 64 bit cuối cùng là độ dài của thông điệp sao cho thỏa điều kiện kích thước thông điệp là một bội số của 512.

*b. Chia thông điệp thành các khối*

Dữ liệu đầu vào của Khối tính toán băm SHA256 là một khối có độ dài 512 bit, vì vậy thông điệp  $M$  sẽ được chia thành  $N$  khối  $M^{(1)}, M^{(2)}, \dots, M^{(n)}$  mỗi khối có độ dài là 512 bit làm đầu vào cho từng khối tính toán băm.

*c. Thiết lập giá trị băm khởi tạo  $H^{(0)}$*

$$H_0^{(0)} = 0x6a09e667$$

$$H_1^{(0)} = 0xbb67ae85$$

$$H_2^{(0)} = 0x3c6ef372$$

$$H_3^{(0)} = 0xa54ff53a$$

$$H_4^{(0)} = 0x510e527f$$

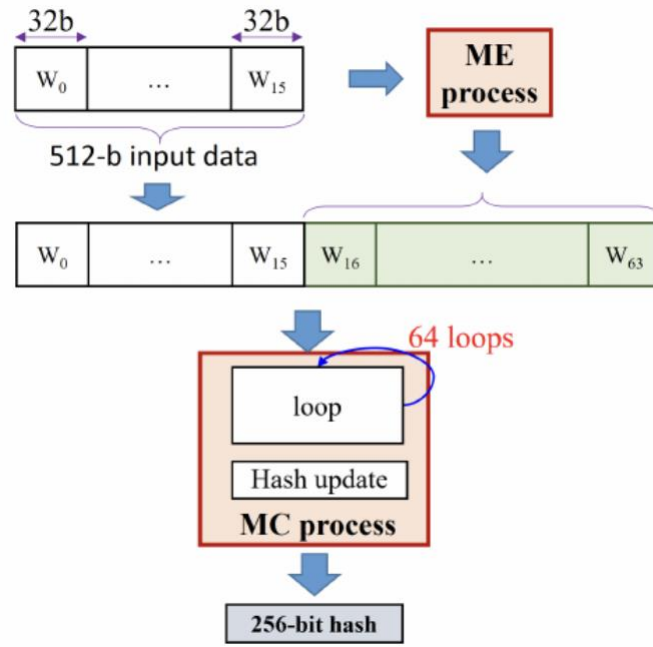
$$H_5^{(0)} = 0x9b05688c$$

$$H_6^{(0)} = 0x1f83d9ab$$

$$H_7^{(0)} = 0x5be0cd19$$

Các hằng số được đại diện cho 32 bit đầu tiên của phần phân số của căn bậc hai của 8 số nguyên tố đầu tiên: **2, 3, 5, 7, 11, 13, 17, 19.**

#### 4.3.2. Tính toán băm SHA256



Hình 4.2: Khối tính toán băm SHA256

Hình 4.2 biểu diễn khối tính toán băm SHA256, nó bao gồm 2 tiến trình chính gọi là Message Expander (ME) và message compressor (CE)

Quá trình ME: khối 512 bit đầu vào mở rộng thành 64 khối dữ liệu  $W_j$  (với  $0 \leq j \leq 63$ ) với mỗi khối  $W_j$  là 1 word (32 bit). Trong 16 vòng đầu tiên, khối 512 bit được gán vào 16 khối từ  $W_0$  đến  $W_{15}$ , 48 khối  $W$  tiếp theo sẽ tính như bên dưới:

$$W_t = \begin{cases} M_t^{(i)} & 0 \leq t \leq 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \leq t \leq 63 \end{cases}$$

Với

$$\begin{aligned} \sigma_0^{\{256\}}(x) &= ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x) \\ \sigma_1^{\{256\}}(x) &= ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x) \end{aligned}$$

Quá trình MC: tính toán giá trị băm 256 bit từ các đầu ra của quá trình ME. Quá trình này gồm 2 bước chính: Vòng lặp loop và cập nhật băm:

- Vòng lặp loop:

+ 8 giá trị băm của vòng lặp ( a, b, c, d, e, f, g, h ) được khởi tạo các giá trị ban đầu  $H_0, H_1, \dots, H_7$ :

$$a = H_0^{(i-1)}$$

$$b = H_1^{(i-1)}$$

$$c = H_2^{(i-1)}$$

$$d = H_3^{(i-1)}$$

$$e = H_4^{(i-1)}$$

$$f = H_5^{(i-1)}$$

$$g = H_6^{(i-1)}$$

$$h = H_7^{(i-1)}$$

+ Các giá trị băm của vòng lặp a, b, c, d, e, f, g, h được tính toán và cập nhật qua 64 vòng lặp:

For  $t=0$  to 63:

{

$$T_1 = h + \sum_1^{\{256\}} (e) + Ch(e, f, g) + K_t^{\{256\}} + W_t$$

$$T_2 = \sum_0^{\{256\}} (a) + Maj(a, b, c)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

}



Với

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

- Cập nhật hàm băm: giá trị băm 256 bit được chia thành 8 khối dữ liệu 32 bit  $H_0^{(i)}$ ,  $H_1^{(i)}, \dots, H_7^{(i)}$  được tính bằng cách cộng các giá trị băm ban đầu  $H_0^{(i-1)}, H_1^{(i-1)}, \dots, H_7^{(i-1)}$  với giá trị băm a, b, c, d, e, f, g, h sau quá trình loop:

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = b + H_1^{(i-1)}$$

$$H_2^{(i)} = c + H_2^{(i-1)}$$

$$H_3^{(i)} = d + H_3^{(i-1)}$$

$$H_4^{(i)} = e + H_4^{(i-1)}$$

$$H_5^{(i)} = f + H_5^{(i-1)}$$

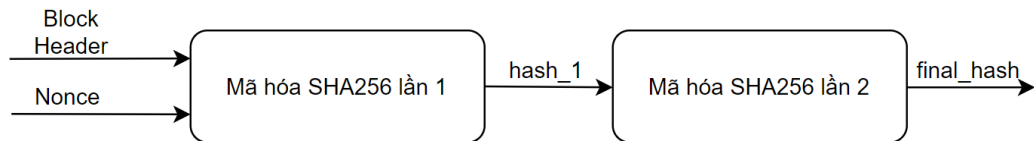
$$H_6^{(i)} = g + H_6^{(i-1)}$$

$$H_7^{(i)} = h + H_7^{(i-1)}$$

Sau khi lặp hết N khối và thực hiện tính toán, giá trị băm cuối cùng của thông điệp M thu được bằng cách ghép các các khối  $H_0^{(N)}, H_1^{(N)}, \dots, H_7^{(N)}$  lại với nhau:

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$$

#### 4.4. Ứng dụng thuật toán SHA256 trong Mining mã hóa Block Header



Hình 4.3: Quá trình mã hóa Block Header

Sau khi Mining Asic nhận Block Header từ Mining Software, Asic tiến hành mã hóa Block Header, quá trình mã hóa Block Header sử dụng thuật toán mã hóa SHA256 trải qua 2 lần mã hóa:

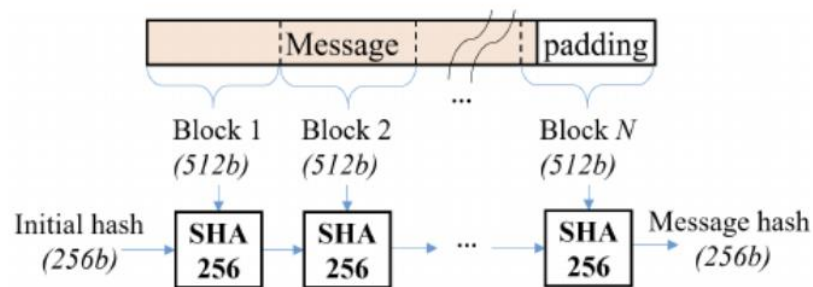
Lần thứ nhất: Block Header (80 byte) được padding tạo thành 2 khối dữ liệu 512 bit, 2 khối dữ liệu này sẽ là đầu vào của Khối mã hóa SHA256

Lần thứ hai: Sau lần mã hóa block header thứ nhất, ta thu được mã băm hash\_1 256 bit, hash\_1 chính là thông điệp đầu vào của lần mã hóa thứ hai, quá trình padding lần 2 diễn ra tuy nhiên lần này chỉ tạo thành 1 khối dữ liệu 512 bit làm đầu vào cho khối SHA256

Sau 2 lần mã hóa, mã băm cuối cùng chính là mã băm được dùng để so sánh với Target Threshold và xác định xem Block Header có hợp lệ hay không. Nếu không hợp lệ thì bắt đầu lại quá trình mã hóa block header kèm số nonce mới.

#### 4.5. IP core SHA256

##### 4.5.1. Đặc tả thiết kế



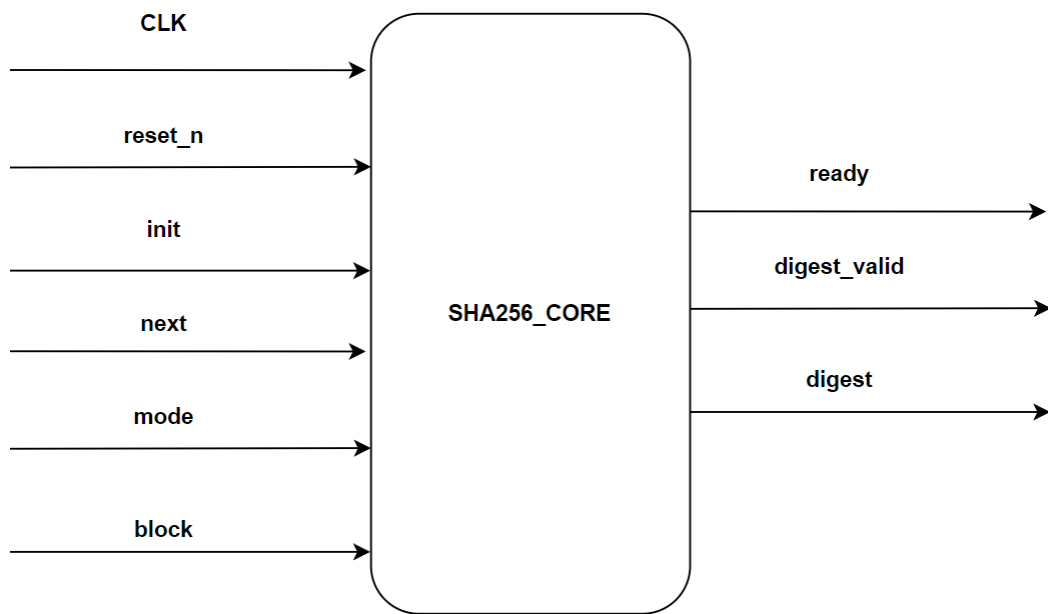
Hình 4.4: Tổng quan thuật toán SHA256

Chuỗi dữ liệu ngõ vào để tiến hành quá trình mã hóa SHA256 có độ dài bất kỳ. Nhưng khối SHA256 chỉ có thể nhận vào một lần là 512 bit để xử lý. Do đó, chuỗi dữ liệu sẽ được chia thành những đoạn 512 bit để đưa tuần tự vào trong khối SHA256. Nếu đoạn dữ liệu cuối cùng không đủ 512 bit thì sẽ được tiến hành padding cho đủ 512 bit.

Các khối dữ liệu 512 bit được đưa vào khối SHA256 để thực hiện mã hóa, các giá trị đầu ra của khối SHA256 chính là một phần giá trị đầu vào cho khối tính toán SHA256 tiếp theo và cứ thực hiện như vậy cho đến khi đi qua hết tất cả các khối dữ liệu, giá trị cuối cùng là mã băm của khối tính toán SHA256 cuối cùng của chuỗi dữ liệu.

#### 4.5.2. Sơ đồ khối SHA256\_Core

Khối SHA256\_Core là khối tính toán SHA256, dữ liệu đầu vào của khối chính là các khối 512 bit từ thông điệp, mã băm 256 bit thu được sau quá trình tính toán ở chân digest. Sơ đồ khối của SHA256\_Core được biểu diễn ở hình 4.4 dưới:



Hình 4.5: Top – view của khối SHA256\_Core

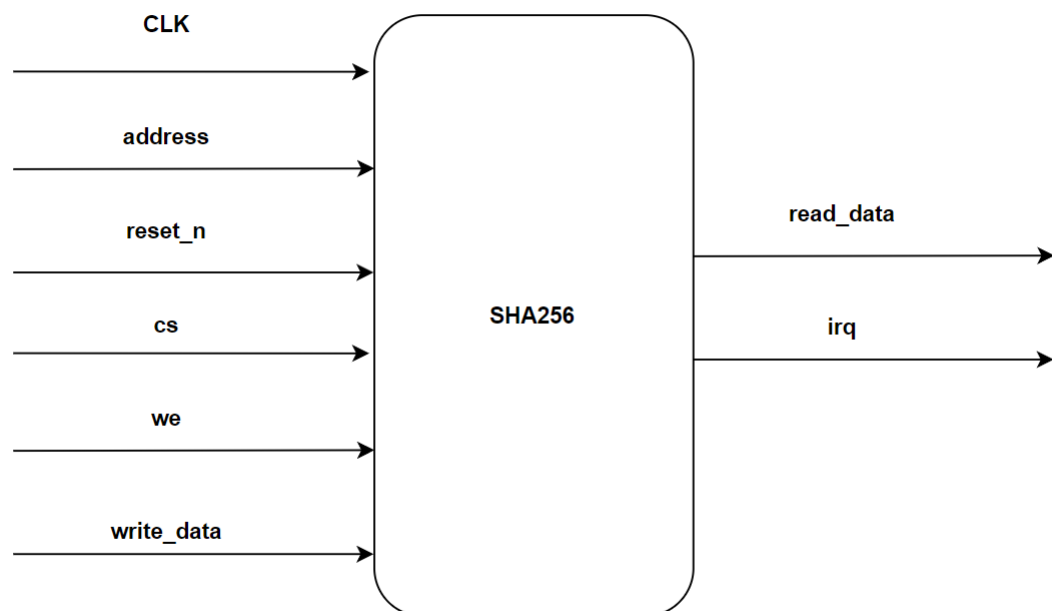
Bảng 4.1: Các tín hiệu của khối SHA256\_Core

STT	Tên tín hiệu	Độ rộng	Hướng	Mô tả
1	clk	1	Input	Tín hiệu clock
2	reset_n	1	Input	Thiết lập lại các tín hiệu
3	init	1	Input	init = 1, khởi tạo các tín hiệu cho quá trình băm của khối đầu tiên
4	next	1	Input	next = 1, khởi tạo các tín hiệu cho quá trình băm của khối tiếp theo
5	mode	1	Input	mode = 1, loại băm SHA256 bit
6	block	512	Input	Khối dữ liệu 512 bit
7	ready	1	Output	ready = 1, sẵn sàng để khởi

				tạo quá trình băm
8	digest_valid	1	Output	digest_valid = 1, mã băm hợp lệ
9	digest	256	Output	Mã băm của khối SHA256

#### 4.5.3. Sơ đồ khối SHA256

Trong đề tài này sử dụng board DE10 Stander sử dụng Avalon Bus để giao tiếp, do đó khối SHA256 được thiết kế để phù hợp với giao tiếp Avalon Bus. Sơ đồ khối của khối SHA256 như hình 4.5:



Hình 4.6: Top-view của khối SHA256

Bảng 4.1: Các tín hiệu của khối SHA256

STT	Tên tín hiệu	Độ rộng	Hướng	Mô tả
1	clk	1	Input	Tín hiệu clock
2	reset_n	1	Input	Tín hiệu reset
3	address	8	Input	Tín hiệu địa chỉ
4	cs	1	Input	cs = 1, bật khối SHA256 hoạt động, ngược lại thì cs = 0
5	we	1	Input	we = 1, cho phép ghi dữ liệu we = 0, đọc dữ liệu
6	write_data	32	Input	Tín hiệu để ghi dữ liệu vào theo word
7	read_data	32	Output	Tín hiệu đọc dữ liệu ra theo word
8	irq	1	Output	Ngắt

## 4.6. Testbench

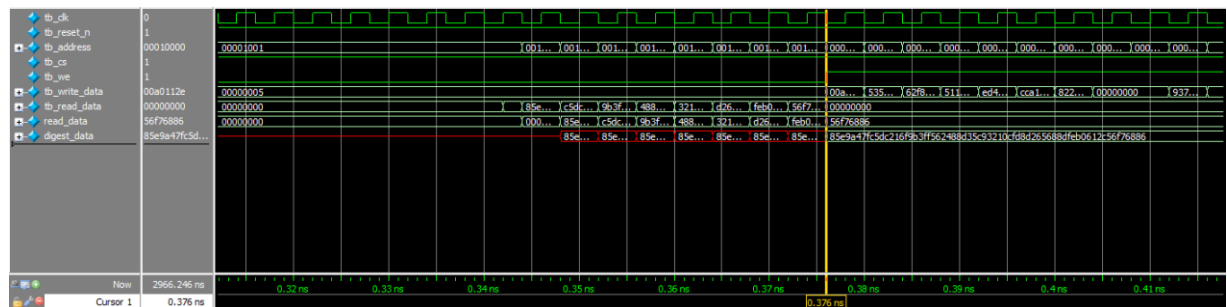
#### 4.6.1. Testbench cho một thông điệp

Testbench thực hiện test cho hai trường hợp:

*Trường hợp thứ nhất:* Thông điệp sau khi dán thêm bit có độ dài chỉ 1 khối 512 bit, khối dữ liệu mẫu sử dụng:

[illegible]

### Kết quả mô phỏng được trên ModelSim



Hình 4.7: Kết quả testbench thông điệp chỉ 1 khối

*Trường hợp thứ hai:* Thông điệp sau khi dán thêm bit có độ dài nhiều khối 512 bit, các khối dữ liệu mẫu sử dụng:

**Khối** **1:**  
512'h00a0112e\_535d5123\_62f80687\_511204ab\_ed45ae64\_cca1a05e\_822a0000\_0000  
0000\_00000000\_937413ff\_6cdb14f4\_8933dbf2\_3ebbe375\_de119214\_347d8a36\_8eac  
954f;

**Khối** **2:**  
512'h94a1141e\_85d82e64\_3e020617\_9448a802\_80000000\_00000000\_00000000\_000  
00000\_00000000\_00000000\_00000000\_00000000\_00000000\_00000000\_00000000\_0  
0000280;

Timing diagram showing digital signals for a system. The signals include db\_clk, db\_reset\_n, db\_address, db\_cs, db\_we, db\_write\_data, db\_read\_data, read\_data, and debug\_data. The diagram shows a sequence of operations with data values like 00001001, 0010..., 0010..., 0010..., 0010..., 0010..., 0010..., 0010..., 00100111, 00000006, 00000000, bc6a2a6a, 0000..., 0f16..., 5584..., 0888..., bc20..., df3a..., 2676..., f199..., bc6a2a6a, 0f16950a358436f8088892fa4bc20e126af3abc02676200ff10912afb56a2a6a. The time scale is 1.07 ns to 1.16 ns.

#### 4.6.2. Testbench mô phỏng quá trình Mining

### Block header:

**Target Threshold:**

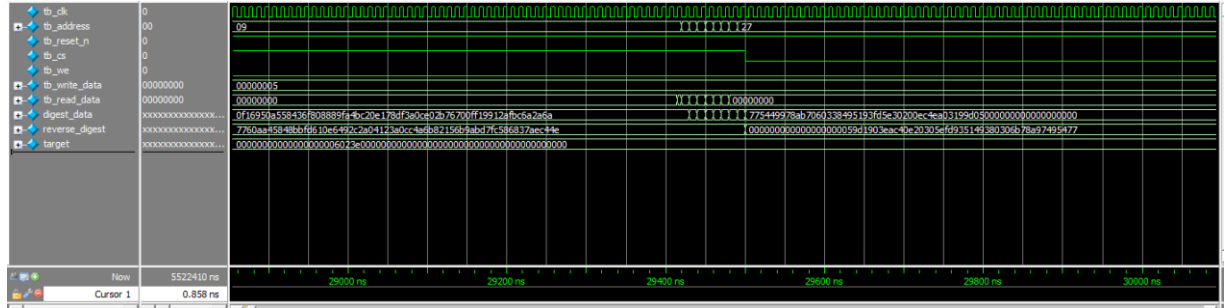
### Kết quả thực hiện trên ModelSim:

[illegible]

---

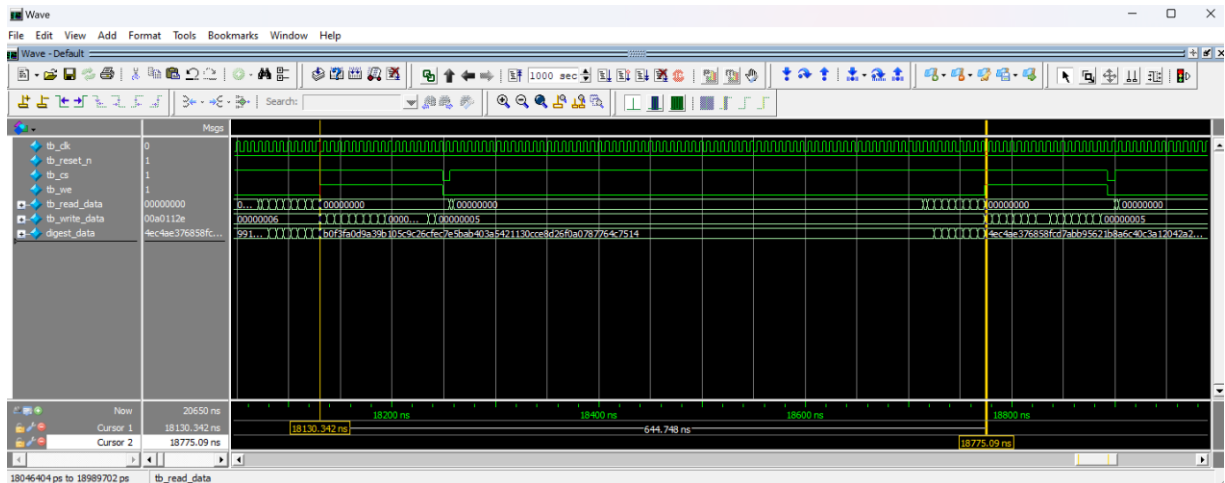
40 | Page





Hình 4.10: Kết quả testbench quá trình Mining (wareform)

#### 4.7. Đánh giá IP core SHA256



Hình 4.11: Tốc độ băm của SHA256

Lõi SHA256 sử dụng tần số 75Mhz, thời gian đo được giữa hai mã băm là 644ns.

$$\text{Băng thông} = \frac{512}{T} = \frac{512}{644 \times 10^{-9}} = 795 \times 10^6 (\text{bit/s}) = \frac{795}{8} \times 10^6 (\text{byte/s})$$

$$= 99 \text{ Mbyte/s}$$

## **Chương 5: Xây dựng hệ thống Mining Asic trên FPGA**

### **5.1. Tổng quan về DE10 Stander FPGA**

Công nghệ FPGA (Field-Programmable Gate Array) là một công nghệ linh hoạt và mạnh mẽ trong lĩnh vực phần cứng, cho phép người dùng tùy chỉnh và thiết kế các mạch logic số theo ý muốn mà không cần phải sử dụng phương pháp thiết kế phần cứng truyền thống. Một trong những ưu điểm chính của công nghệ FPGA là khả năng lập trình lại và tái cấu hình.

DE10 Stander là một bo mạch FPGA (Field-Programmable Gate Array) được phát triển bởi Terasic Technologies. Được xây dựng trên cơ sở chip Intel Cyclone V, DE10-Standard cung cấp một nền tảng phát triển linh hoạt cho các ứng dụng điện tử, hệ thống nhúng, và phát triển các hệ thống số.

Một số tính năng quan trọng và cổng kết nối đa dạng, bao gồm:

- + Chip FPGA Cyclone V: Chip cung cấp khả năng lập trình linh hoạt, cho phép tùy chỉnh và triển khai các hệ thống số.
- + Bộ nhớ DDR3: Bo mạch được trang bị bộ nhớ DDR3 tích hợp, cung cấp khả năng xử lý dữ liệu nhanh chóng và lưu trữ dữ liệu lớn.
- + Cổng giao tiếp đa dạng: cổng HDMI, cổng VGA, cổng âm thanh, cổng Ethernet, cổng USB và các cổng GPIO giúp kết nối và tương tác với các thiết bị ngoại vi và mạng khác nhau.
- + Khe cắm thẻ nhớ SD: một khe cắm thẻ nhớ SD để dễ dàng truy cập và lưu trữ dữ liệu.
- + Bộ điều khiển ARM Cortex-A9: DE10-Standard tích hợp một bộ vi xử lý ARM Cortex-A9, cho phép chạy các ứng dụng trên hệ điều hành Linux hoặc hệ điều hành nhúng.

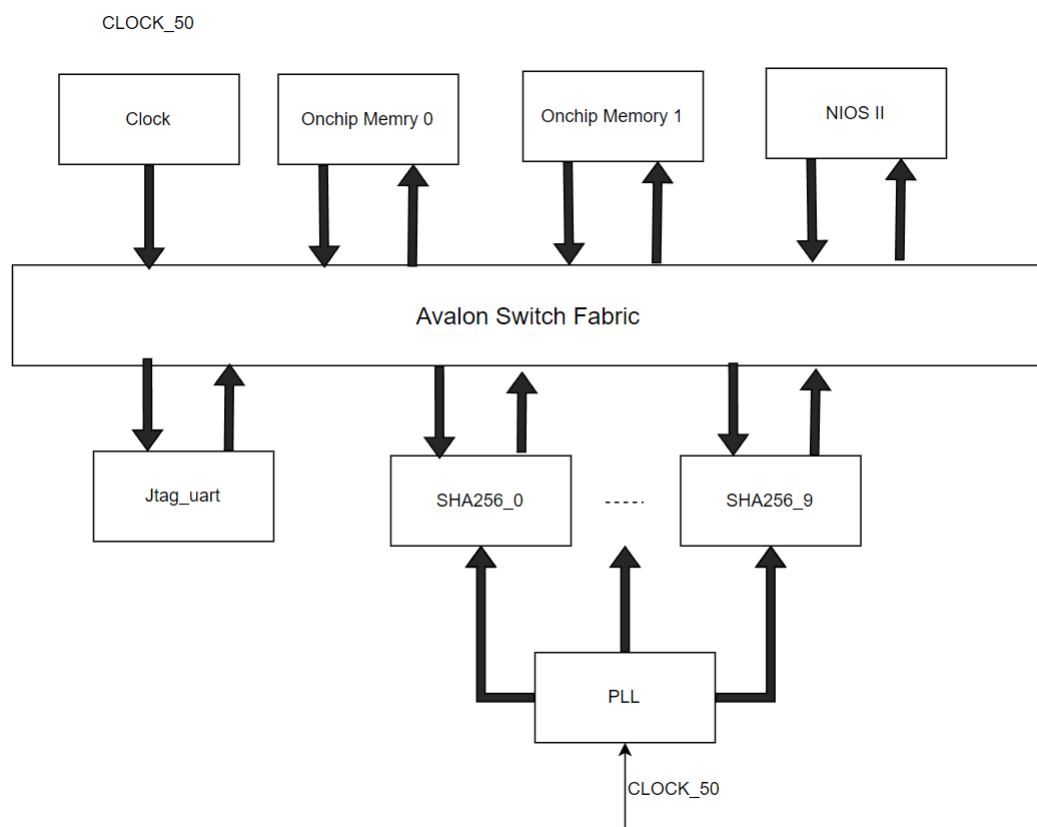
- + Các công nghệ giao tiếp khác: Ngoài các cổng kết nối cơ bản, DE10-Standard hỗ trợ các công nghệ giao tiếp khác như Bluetooth và Wi-Fi thông qua các mô-đun mở rộng tùy chọn.

## 5.2. Thiết kế phần cứng trên Platform Designer

Hệ thống gồm:

Thành phần	Mô tả
CLOCK_50	Sử dụng clock 50MHz cho hệ thống
NIOS II	Vi xử lý NIOS II
Jtag_uart	Truyền nhận dữ liệu nối tiếp
OnChip Memory 0	Bộ nhớ sử dụng với mục đích lưu trữ lệnh
OnChip Memory 1	Bộ nhớ sử dụng với mục đích lưu trữ dữ liệu trong quá trình mining
PLL	Tạo Xung 75MHz cung cấp cho clock cho core SHA256
SHA256_0 đến SHA256_9	10 Core SHA256

Sơ đồ tổng quan hệ thống:



Hình 5.1: Tổng quan hệ thống

Hệ thống được xây dựng trên Platform Designer, chi tiết hệ thống:

Connections	Name	Description	Export	Clock	Base	End	Tags	Opcode Name
	<b>system_clk</b>	Clock Source						
	clk_in	Clock Input	sys_clk	exported				
	clk_in_reset	Reset Input	reset	system_clk				
	clk_reset	Clock Output	Double-click					
	clk_reset	Reset Output	Double-click					
	<b>nios2_gen2_0</b>	Nios II Processor						
	clk	Clock Input	Double-click	system_clk				
	reset	Reset Input	Double-click	[clk]				
	data_master	Avalon Memory Ma...	Double-click	[clk]				
	instruction_master	Avalon Memory Ma...	Double-click	[clk]				
	irq	Interrupt Receiver	Double-click	[clk]				
	debug_reset_request	Reset Output	Double-click	[clk]				
	debug_mem_slave	Avalon Memory Ma...	Double-click	[clk]				
	custom_instruction_master	Custom Instruction...	Double-click	[clk]				
	<b>onchip_memory2_0</b>	On-Chip Memory (...)						
	clk1	Clock Input	Double-click	system_clk				
	s1	Avalon Memory Ma...	Double-click	[clk1]				
	reset1	Reset Input	Double-click	[clk1]				
	<b>jtag_uart_0</b>	JTAG UART Intel F...						
	clk	Clock Input	Double-click	system_clk				
	reset	Reset Input	Double-click	[clk]				
	avalon_jtag_slave	Avalon Memory Ma...	Double-click	[clk]				
	irq	Interrupt Sender	Double-click	[clk]				
	<b>pll_0</b>	PLL Intel FPGA IP						
	refclk	Clock Input	Double-click	system_clk				
	reset	Reset Input	Double-click	[clk]				
	outclk0	Clock Output	Double-click	pll_0_outclk0				
	locked	Conduit	Double-click					
	<b>SHA_0</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_1</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_2</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_3</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_4</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_5</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_6</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_7</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_8</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>SHA_9</b>	SHA_2						
	clock	Clock Input	Double-click	pll_0_outclk0				
	reset	Reset Input	Double-click	[clock]				
	avalon_slave_0	Avalon Memory Ma...	Double-click	[clock]				
	interrupt_sender	Interrupt Sender	Double-click	[clock]				
	<b>sdram</b>	SDRAM Controller ...						
	clk	Clock Input	Double-click	system_clk				
	reset	Reset Input	Double-click	[clk]				
	s1	Avalon Memory Ma...	Double-click	[clk]				
	wire	Conduit	Double-click	sdram_wire				
	<b>onchip_memory2_1</b>	On-Chip Memory (...)						
	clk1	Clock Input	Double-click	system_clk				
	s1	Avalon Memory Ma...	Double-click	[clk1]				
	reset1	Reset Input	Double-click	[clk1]				

Hình 5.2: Chi tiết hệ thống trên Platform Designer

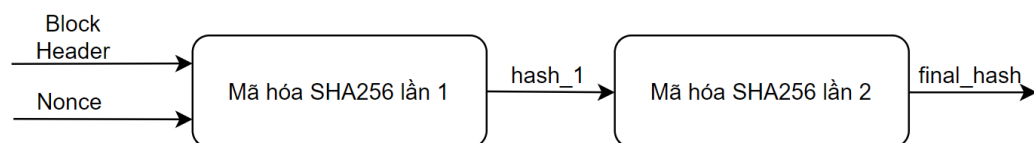
Kết quả phân tích hệ thống trên Quartus:

Flow Summary	
<<Filter>>	
Flow Status	Successful - Wed Jul 05 18:14:19 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	Asic_Miner
Top-level Entity Name	Asic_Miner
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	15,554 / 41,910 ( 37 % )
Total registers	29138
Total pins	41 / 499 ( 8 % )
Total virtual pins	0
Total block memory bits	4,148,864 / 5,662,720 ( 73 % )
Total DSP Blocks	0 / 112 ( 0 % )
Total HSSI RX PCSs	0 / 9 ( 0 % )
Total HSSI PMA RX Deserializers	0 / 9 ( 0 % )
Total HSSI TX PCSs	0 / 9 ( 0 % )
Total HSSI PMA TX Serializers	0 / 9 ( 0 % )
Total PLLs	1 / 15 ( 7 % )
Total DLLs	0 / 4 ( 0 % )

Hình 5.3: Tổng hợp tài nguyên đã sử dụng

### 5.3. Xây dựng phần mềm

#### 5.3.1. Phân chia bộ nhớ cho mỗi khối tính toán SHA256:



Hình 5.4: Quá trình mã hóa Block Header

Như đã đề cập ở trên, quá trình mã hóa Block Header sử dụng thuật toán SHA256 phải qua 2 lần mã hóa. Ở lần thứ nhất, Block Header 80 byte được padding tạo thành 2 khối dữ liệu 512 bit. Ở lần mã hóa thứ hai, hash\_1 (256 bit) được padding tạo thành 1 khối dữ liệu 512 bit. Như vậy vùng nhớ tối thiểu để thực hiện lưu trữ và tính toán cho mỗi core SHA trong bộ nhớ là  $512 + 512 + 512 = 1536 \text{ bit} = 192 \text{ byte} = 48 \text{ byte}$ .

Trong thiết kế sử dụng 10 core SHA256, như vậy chúng ta có thể phân chia bộ nhớ như sau:

Onchip Memory 1	
mem_base	Vùng nhớ lưu trữ và tính toán cho Core 0
mem_base + 48	Vùng nhớ lưu trữ và tính toán cho Core 1
mem_base + 48 * 2	Vùng nhớ lưu trữ và tính toán cho Core 2
mem_base + 48 * 3	Vùng nhớ lưu trữ và tính toán cho Core 3
...	
mem_base + 48 * 8	Vùng nhớ lưu trữ và tính toán cho Core 8
mem_base + 48 * 9	Vùng nhớ lưu trữ và tính toán cho Core 9
mem_base + 48 * 10	Vùng nhớ lưu trữ Target Threshold

Hình 5.5: Phân chia bộ nhớ cho các core SHA256 trong bộ nhớ

### 5.3.2. Xây dựng phần mềm:

Phần mềm được xây dựng trong Eclipse tích hợp sẵn trong Quartus. Flowchart biểu diễn luồng đi của dữ liệu như hình 5.2.

Theo như cơ chế Mining đã trình bày ở chương 3, Mining Asic thực hiện quá trình thử và sai, tìm kiếm số nonce đúng bắt đầu từ 0x00000000 đến 0xffffffff. Thay vì thực hiện như vậy, chúng ta sẽ thực hiện chia 4 byte nonce đó ra 10 khoảng cách đều nhau:

**unsigned int** nonce0 = 0x00000000;

**unsigned int** nonce1 = 0x19999999;

**unsigned int** nonce2 = 0x33333333;

**unsigned int** nonce3 = 0x4ccccccc

**unsigned int** nonce4 = 0x66666666;

**unsigned int** nonce5 = 0x7ffffff;

**unsigned int** nonce6 = 0x93333333;

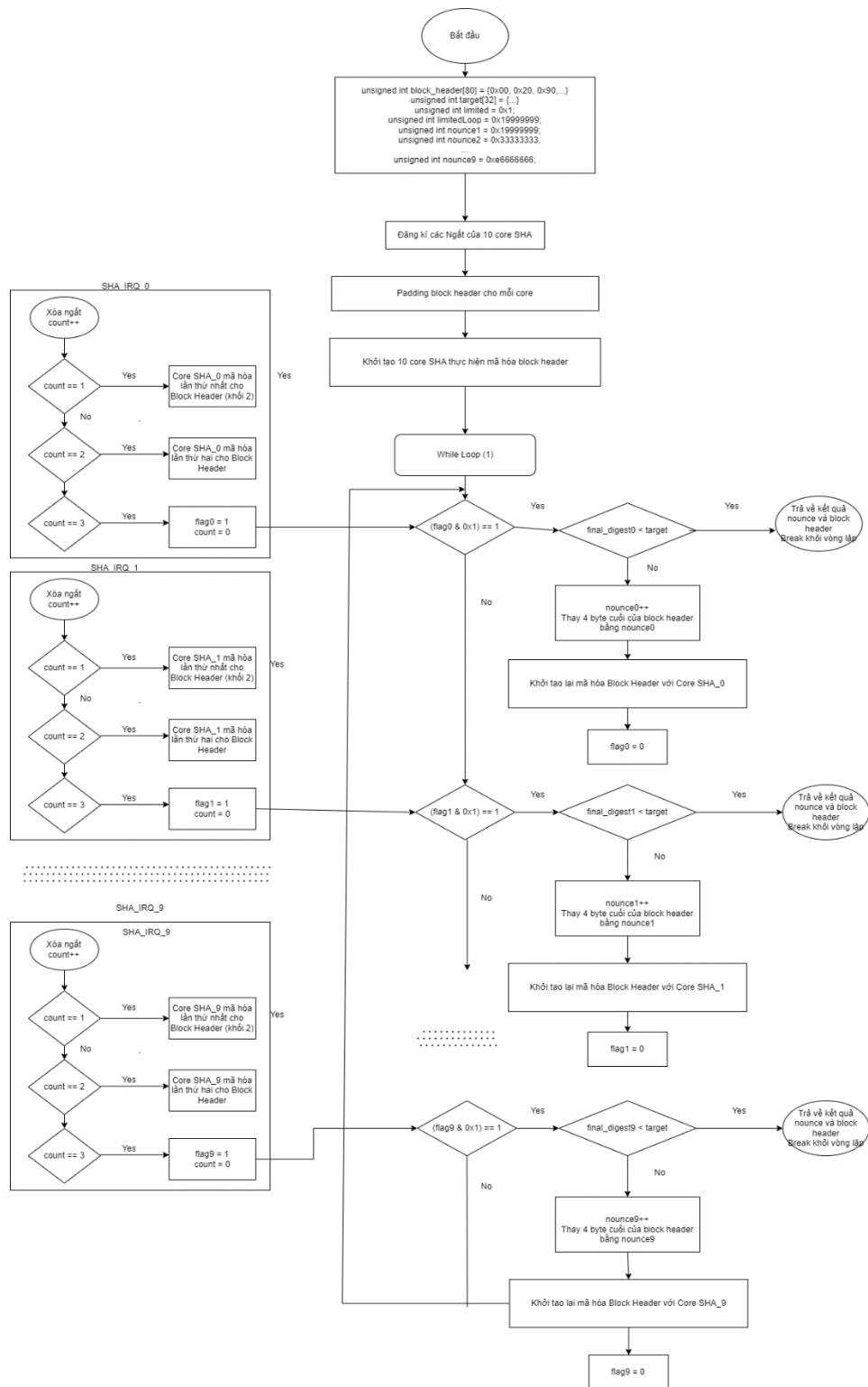
**unsigned int** nonce7 = 0xb3333333;

**unsigned int** nonce8 = 0xcccccccc;

**unsigned int** nonce9 = 0xe6666666;

Mỗi core SHA256 sẽ đảm nhiệm việc mã hóa Block Header với khoảng nonce riêng, như vậy hệ thống sẽ tăng tốc độ và khả năng tìm kiếm ra block header hợp lệ.





Hình 5.6: Flowchart quá trình mining

## Chương 6: Thử nghiệm và đánh giá

### 6.1. Mục tiêu thử nghiệm

Hệ thống Mining Asic được thử nghiệm với các mục tiêu chính:

- Theo dõi sự ổn định của hệ thống khi hoạt động.
- Tính chính xác khi mã hóa block header sử dụng lõi SHA256.
- Theo dõi và đánh giá tính chính xác của hệ thống Mining, đối chiếu kết quả Mining được với Block Header hợp lệ đã được chấp nhận trên mạng Blockchain.
- Theo dõi và đánh giá tốc độ của hệ thống.

### 6.2. Thiết lập thử nghiệm

Hệ thống được xây dựng và thực hiện trên board DE10 stander FPGA, chúng ta sẽ tiến hành thiết kế phần cứng trên Platform Designer và tích hợp hệ thống vào trong project Quartus, tiếp theo chúng ta sẽ xây dựng phần mềm theo flow đã trình bày ở trên.

Chúng ta sẽ thiết lập 2 trường hợp cho thử nghiệm:

*Trường hợp 1:* Thực hiện quá trình Mining chỉ với 1 core SHA256

*Trường hợp 2:* Thực hiện quá trình Mining với 10 core SHA256

Trong mỗi trường hợp, chúng ta sẽ thiết lập 3 thử nghiệm cho hệ thống tương ứng với việc mining 3 Block Header mẫu:

#### **Block Header 1 :**

```
00a0112e535d512362f80687511204abed45ae64cca1a05e822a000000000000000000  
0937413ff6cdb14f48933dbf23ebbe375de119214347d8a368eac954f94a1141e85d82e64  
3e02061700000000
```

**Block Header 2 :**

010000009810f0fa1817a4d2d371a069addaafab2ca99887abcc5bd2528e434100000000  
654f005a6e4b4b57b42343fb0e47f32079b4ebfe643c2ea4ea20e46c3af00c238d466849ff  
ff001d000000000

**Block Header 3 :**

002090209540d4bbb6512d257236f1a68176c81a2a9495e0258001000000000000000000  
04e0285b59d7c572637d9fe928d31dba237bee2e1b03e53cf0283acd27ede72ab4bfa386  
4b2e005174859c1e5

**6.3. Kết quả thử nghiệm****a. Theo dõi sự ổn định của hệ thống khi hoạt động:**

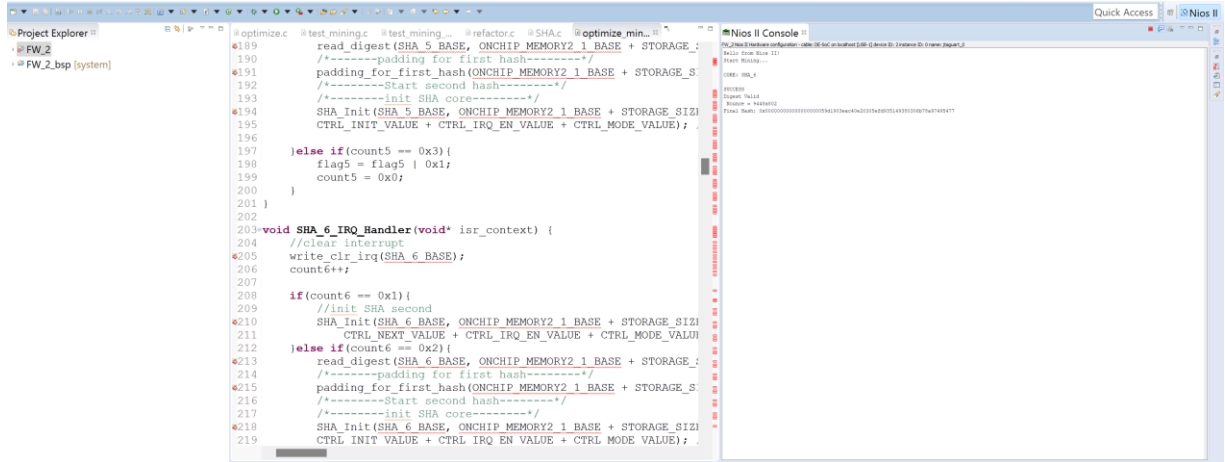
Hệ thống hoạt động ổn định khi thực hiện việc mining, theo dõi hệ thống hoạt động vài giờ liên tục mà không hề có sự cố gây gián đoạn việc mining trên thiết bị.

**b. Tính chính xác khi mã hóa block header sử dụng lõi SHA256**

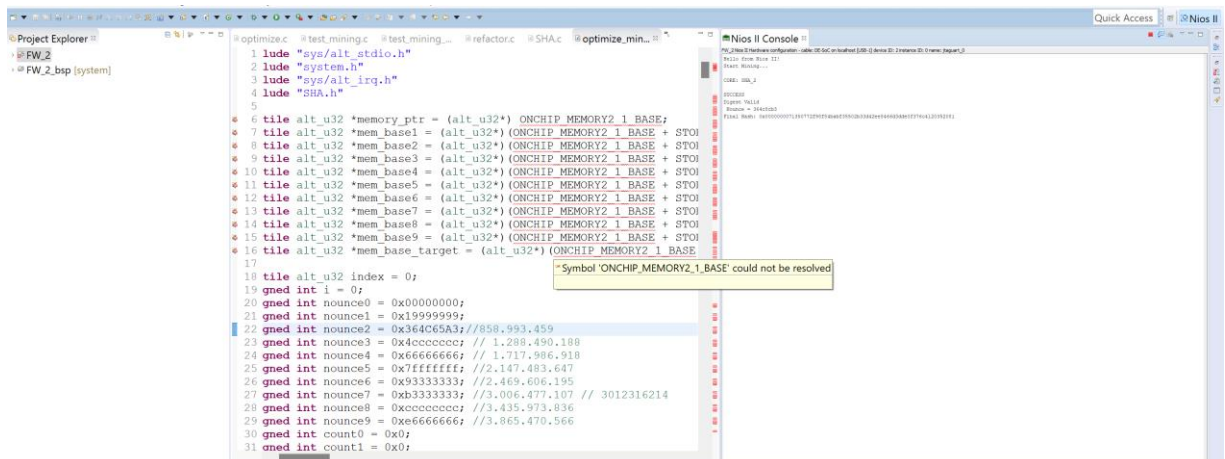
Trong quá trình hệ thống thực hiện mining, theo dõi các mã hóa block header thông qua uart\_jtag cho ra kết quả mã hóa chính xác.

**c. Theo dõi và đánh giá tính chính xác của hệ thống Mining, đối chiếu kết quả Mining được với Block Header hợp lệ đã được chấp nhận trên mạng Blockchain**

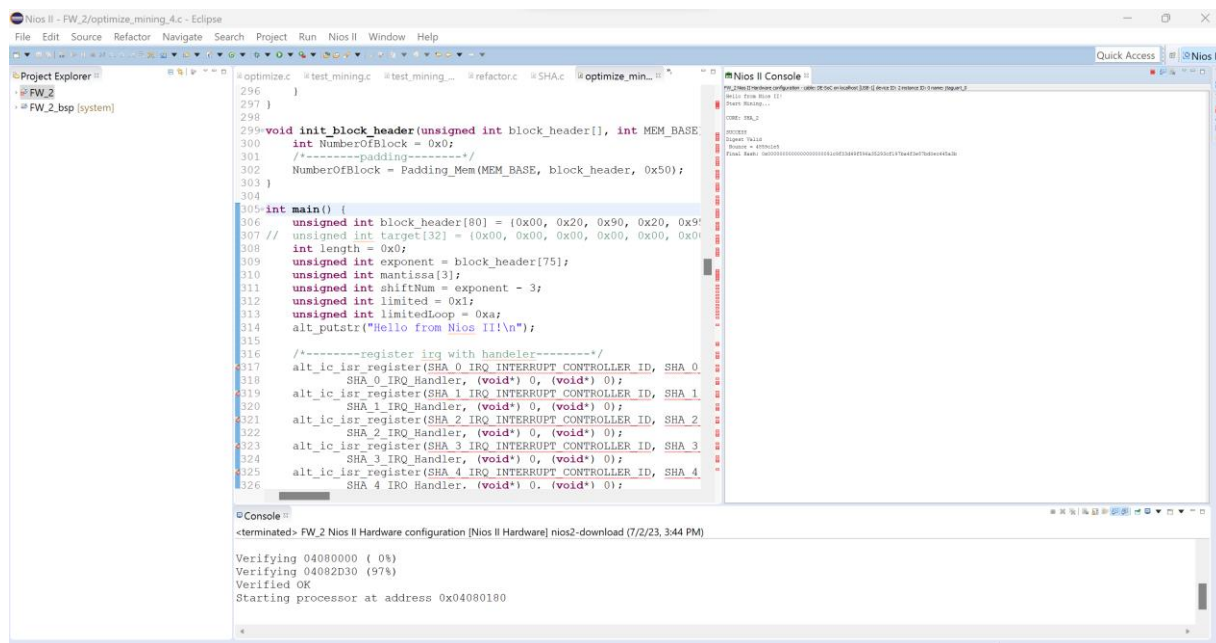
Quá trình mining cho 3 block header mẫu cho ra kết quả như sau:



Hình 6.1: Kết quả mining cho Block Header 1



Hình 6.2: Kết quả mining cho Block Header 2



Hình 6.3: Kết quả mining cho Block Header 3


Ta thấy kết quả mining cho ra 3 giá trị băm của 3 block header hợp lệ lần lượt là:


Hash 1: 0x00000000000000000000000059d1903eac40e20305efd935149380306b78a97495477

Hash 2: 0x0000000071350772f98f84babf35502b33d42ee8466d3dde0f376c4120352081

Hash 3: 0x00000000000000000000000091c9f33d49f596a35293cf197ba4f3e87bd0ec445a3b


Đối chiếu khớp với block đã được thêm blockchain:


[Blockchains](#)
[FAQ](#)
[API](#)




# Bitcoin Blockchain Explorer


up to block 797381



[the most popular place to buy, sell and accept Bitcoin](#)

## BTC block height #784217

Hash	< 0000000000000000059d1903eac40e20305efd935149380306b78a97495477 > 
Date/Time	4/6/2023, 9:34:45 PM (UTC+7:00)
Transactions	2723 <span>1.33 MB</span>
Value Out	11,639.87398508 BTC
Difficulty	46843400286276.5
Outstanding	19,338,323.9615688 BTC
Created	6.25000000 BTC


Hình 6.4: Block 1 trên mạng blockchain (bitcoin)


[Blockchains](#)
[FAQ](#)
[API](#)




# Bitcoin Blockchain Explorer

up to block 797381


[the most popular place to buy, sell and accept Bitcoin](#)

## BTC block height #27

Hash	< 0000000071350772f98f84babf35502b33d42ee8466d3dde0f376c4120352081 > 
Date/Time	1/10/2009, 1:56:13 PM (UTC+7:00)
Transactions	1 <span>0.2 kB</span>
Value Out	50.0 BTC
Difficulty	1
Outstanding	1,350.0 BTC
Created	50.0 BTC

Hình 6.5: Block 2 trên mạng Blockchain (bitcoin)



Đánh giá:

Số chu kì cho một lần hash: 18

Thời gian cho một lần hash :  $T = \frac{1}{50 \times 10^6} \times 18 = 36 \times 10^{-8} \text{ (s)}$

Băng thông:  $B = \frac{512}{T} = \frac{512}{36 \times 10^{-8}} = 14.2 \times 10^8 \text{ bit/s} = \frac{14.2 \times 10^8}{8 \times 1024 \times 1024} \text{ (byte/s)} = 169$

Mbyte/s

Khảo sát tốc độ mining của hệ thống

Bảng 6.1: Khảo sát tốc độ của hệ thống Mining

	<b>1 core SHA256</b>	<b>10 core SHA256</b>
1000 nounce	Ngay lập tức	1.4 s
10.000 nounce	1.4 s	14s
100.000 nounce	14s	2p18s

Đánh giá:

- Tốc độ hash block header trong 1 giây của hệ thống khoảng 7142 số nonce
- 1 block header khi mã hóa thì phải qua khối SHA256 3 lần, vậy tốc độ hash của hệ thống:  $7142 \times 3 = 21428 \text{ hash/s} = 21.4 \text{ KH/s}$



#### **6.4. Kết luận và hướng phát triển**

Qua quá trình thử nghiệm cho ta thấy được hệ thống Mining Asic hoạt động ổn định và chính xác. Tốc độ mã hóa block header cho một core SHA256 đạt khoảng 21.5 KH/s. Tuy nhiên hệ thống vẫn chưa được tối ưu về phần mềm nên khi hoạt động với nhiều core SHA256 vẫn chưa đạt được tốc độ theo ý muốn.

*Hướng phát triển :*

Hệ thống Mining Asic cần được tối ưu thêm về mặt phần mềm để tăng tốc độ hash khi hoạt động với nhiều core SHA256.

Hệ thống đang được xây dựng với 10 core SHA256 và chiếm khoảng 37% các khối logic trên board DE10 Stander, chúng ta có thể mở rộng lên đến 29 core SHA256, khi đó tốc độ có thể đạt được của hệ thống là:  $21.4 \text{ Kh/s} * 29 = 620.6 \text{ Kh/s}$ . Như vậy sẽ tăng tốc độ và xác suất tìm ra được Block Header lên rất nhiều lần.

## **GIẢI THÍCH THUẬT NGỮ**

- (1) **Bitcoin** : Là một phần mềm đồng bộ hóa và quản lý blockchain cho hệ thống mạng lưới Bitcoin. Đây là một nút Bitcoin đầy đủ (full node) và được phát triển bởi nhóm Bitcoin Core. Cung cấp API cho các ứng dụng khác để truy cập và tương tác với blockchain.
- (2) **Mempool**: là một danh sách tất cả các giao dịch hợp lệ mà mỗi node lưu trữ và cập nhật liên tục. Các giao dịch trong mempool chưa được xác nhận bởi bất kỳ khối nào trong blockchain.
- (3) **Mining Pool**: là một nhóm hoặc mạng lưới của các máy đào (miners) hoạt động cùng nhau để tăng hiệu suất đào và chia sẻ phần thưởng, Mining pool cho phép các miner kết hợp sức mạnh tính toán để tăng cơ hội tìm ra block mới và nhận phần thưởng.

## **TÀI LIỆU THAM KHẢO**

- [1] Embedded Peripherals IP User Guide - Intel
- [2] Nios® II Processor Reference Guide - Intel
- [3] Avalon® Interface Specifications - Intel
- [4] DE10-Standard User Manual - IIS Windows Server - Intel
- [5] Embedded Software Architecture - J. A. Cook and J. S. Freudenberg
- [6] Field-Programmable Gate Array Technology - Stephen M. Trimberger.
- [7] FPGA-Based Implementation of Signal Processing Systems - Roger Woods, John McAllister, Gaye Lightbody và Ying Yi.
- [8] Applied Cryptography: Protocols, Algorithms, and Source Code in C - Bruce Schneier.
- [9] Understanding Cryptography: A Textbook for Students and Practitioners - Christof Paar và Jan Pelzl.