

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH



BÁO CÁO TỔNG KẾT
MÔN THIẾT KẾ LUẬN LÝ SỐ – CE118.N22
Đề tài: Thiết kế bộ nhân, chia 32 bits không dấu

Sinh viên thực hiện:
Phùng Đức Bảo 21521860
Trần Triều Trung 21522727

GIẢNG VIÊN HƯỚNG DẪN:
TRƯƠNG VĂN CƯƠNG

TP. HỒ CHÍ MINH – Tháng 05 năm 2023

Mục lục

Phần 1. Mở đầu.....	2
1.1. Đặt vấn đề	2
1.2. Mục đích, yêu cầu	2
1.2.1. Mục đích	2
1.2.2. Yêu cầu	2
1.2.3. Tiến độ thực hiện	2
Phần 2. Nội dung	3
2.1. Phép nhân 32 bits số nguyên không dấu	3
2.1.1. Lưu đồ giải thuật.....	3
2.1.2. Datapath	3
2.1.3. Control	5
2.1.4. Kết quả mô phỏng.....	7
2.1.5. Cải tiến phép nhân có dấu từ phép nhân không dấu	7
2.1.6. Đánh giá kết quả phép nhân.....	8
2.2. Phép chia 32 bits số nguyên không dấu	11
2.2.1. Lưu đồ giải thuật.....	11
2.2.2. Datapath	11
2.2.3. Control	13
2.2.4. Kết quả mô phỏng.....	16
2.2.5. Cải tiến phép chia có dấu từ phép chia không dấu	16
2.2.6. Đánh giá kết quả phép chia.....	17

Danh mục hình ảnh

Hình 1: Lưu đồ giải thuật phép nhân.....	3
Hình 2: Datapath phép nhân.....	4
Hình 3: Khối control phép nhân.....	5
Hình 4: Lưu đồ chuyển trạng thái phép nhân.....	6
Hình 5: kết quả mô phỏng phép nhân.....	7
Hình 6: Mạch chuyển nhân không dấu sang nhân có dấu.....	7
Hình 7: Mạch phép nhân có dấu.....	8
Hình 8: Mô phỏng kết quả phép nhân có dấu.....	8
Hình 9: Đánh giá kết quả phép nhân - 1.....	9
Hình 10: Đánh giá kết quả phép nhân - 2.....	9
Hình 11: Lưu đồ giải thuật phép chia.....	11
Hình 12: Datapath phép chia.....	12
Hình 13: Khối control phép chia.....	13
Hình 14: Lưu đồ chuyển trạng thái phép chia.....	14
Hình 15: kết quả mô phỏng phép chia.....	16
Hình 16: Mạch phép chia có dấu.....	16
Hình 17: Mô phỏng kết quả phép chia có dấu.....	17
Hình 18: Đánh giá kết quả phép chia - 1.....	17
Hình 19: Đánh giá kết quả phép chia - 2.....	18

Danh mục bảng

Bảng 1: Bảng sự thật Shift Register	4
Bảng 2: Bảng sự thật Register	5
Bảng 3: Bảng control word phép nhân	6
Bảng 4: Bảng ngõ ra khỏi control phép nhân	6
Bảng 5: Bảng control word phép chia	14
Bảng 6: Bảng ngõ vào DFF phép chia.....	15
Bảng 7: Bảng ngõ ra khỏi control phép chia	15

Lời nói đầu

Máy tính nói riêng và các thiết bị tính toán nói chung đều sử dụng các phép toán đơn giản là cộng trừ nhân chia, là những phép toán cơ bản nhất và quan trọng nhất. Nhưng chúng ta đã biết rằng máy tính chỉ hiểu được các con số 0 và 1, vậy làm thế nào để chúng có thể thực hiện được các phép toán đó? Từ câu hỏi này, chúng em sẽ dùng các kiến thức đã học trong môn Thiết kế luận lý số để thực hiện 2 phép toán là phép nhân và phép chia 32 bits số nguyên không dấu.

Mặc dù đã cố gắng hết sức nhưng cũng không thể tránh khỏi sai sót về hình thức và nội dung trong đề tài này. Kính mong quý thầy cô và các bạn góp ý để chúng em có thể hoàn thành tốt hơn đề tài này. Xin chân thành cảm ơn.

Phần 1. Mở đầu

1.1. Đặt vấn đề

Các phép toán số học chính là các phép toán cơ bản nhất và quan trọng nhất trong toán học. Nhờ phát minh ra các phép toán này mà cuộc sống con người trở nên thuận tiện hơn rất nhiều. Và con người đã thành công trong việc đưa các phép toán này vào thế giới của máy tính, nơi chỉ có các con số 0 và 1. Hiểu biết được sự quan trọng và hấp dẫn đó, nhóm chúng em sẽ dùng các kiến thức đã học trong môn học Thiết kế luận lý số để thực hiện phép toán nhân và chia 32 bits số nguyên không dấu.

1.2. Mục đích, yêu cầu

1.2.1. Mục đích

Giúp sinh viên và tất cả mọi người những ai quan tâm và hứng thú với việc tìm hiểu kiến thức về máy tính nói chung và kiến thức về luận lý số nói riêng. Bài viết này sẽ trình bày những hiểu biết của nhóm chúng em để mô tả một cách mà phép nhân và chia được thực hiện bằng các thành phần lưu trữ, tính toán, các cổng logic.

1.2.2. Yêu cầu

Giải quyết được các phép tính nhân và chia giới hạn trong khoảng 32 bits số nguyên không dấu. Hiểu rõ các thành phần lưu trữ, tính toán, các phép toán đại số boolean đã được học. Hiểu và có thể thiết kế một số mạch thực hiện các phép toán đơn giản.

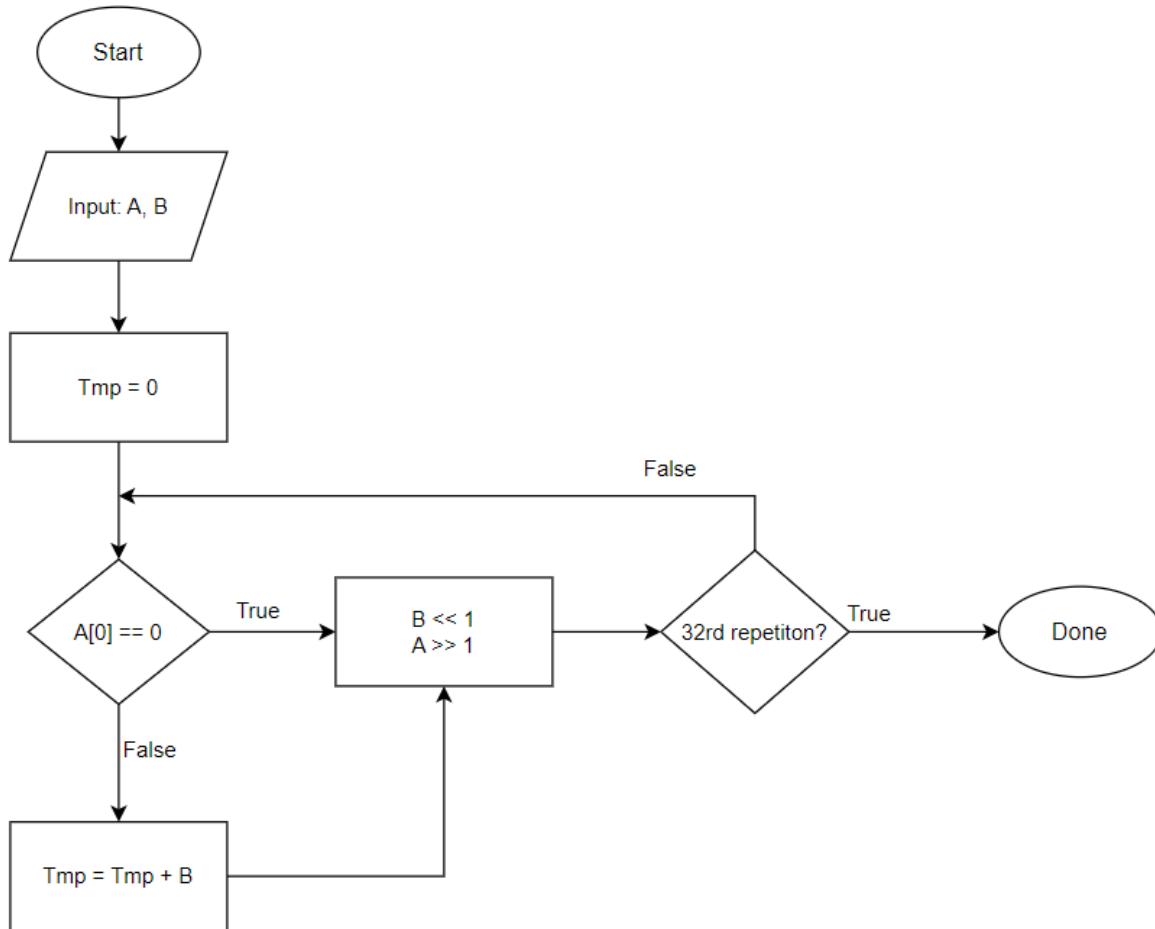
1.2.3. Tiến độ thực hiện

Đã hoàn thành các yêu cầu đặt ra. Kết quả thực hiện được mô phỏng trên phần mềm Quartus cho kết quả đúng yêu cầu của bài toán.

Phần 2. Nội dung

2.1. Phép nhân 32 bits số nguyên không dấu

2.1.1. Lưu đồ giải thuật



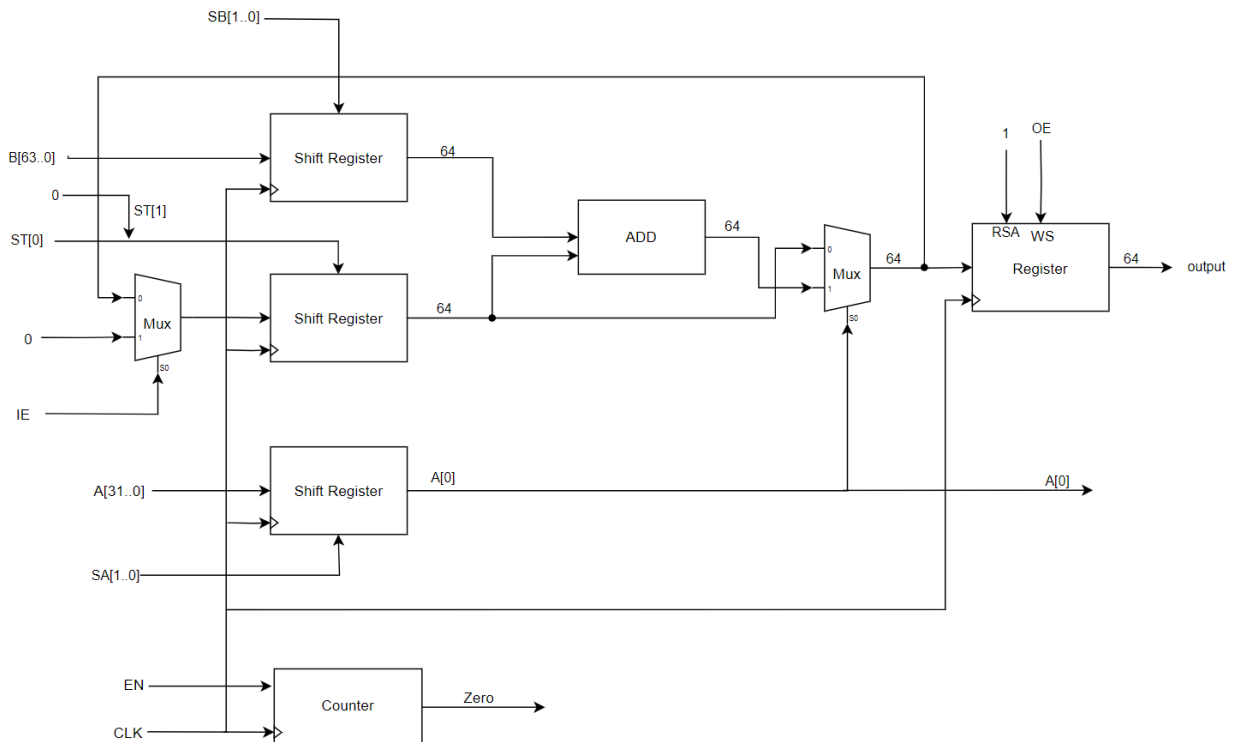
Hình 1: Lưu đồ giải thuật phép nhân

2.1.2. Datapath

Dựa vào lưu đồ giải thuật, đầu tiên ta cần xác định được những thành phần cần thiết để tạo nên datapath

- Thanh ghi dịch để lưu các giá trị A (số nhân), B (số bị nhân), Tmp (kết quả tạm thời) và thực hiện dịch bit
- Bộ counter thực hiện việc đếm cho vòng lặp 32 lần
- Bộ cộng để cộng lấy kết quả tạm thời
- Các mux2to1 64 bits để lựa chọn đầu ra theo điều kiện
- Thanh ghi lưu trữ kết quả cuối cùng

Hình vẽ dưới đây mô tả đường dữ liệu cho phép tính nhân



Hình 2: Datapath phép nhân

- Input:

- + A: 32 bits, lưu số bị nhân
- + SA[1..0]: điều khiển thanh ghi lưu A
- + B: 64bits, lưu số nhân
- + SB[1..0]: điều khiển thanh ghi lưu B
- + T: 64bits, lưu Tmp
- + ST: 1 bits, điều khiển thanh ghi lưu Tmp
- + EN: điều khiển hoạt động của counter
- + OE: tín hiệu cho phép xuất kết quả

- Output:

- + Zero: tín hiệu của counter khi lặp đủ 32 lần
- + A[0]: tín hiệu điều kiện thực hiện phép tính cộng hoặc bỏ qua
- + output: ngõ ra kết quả cho phép nhân

Bảng sự thật Shift register

Ngõ vào điều khiển S	Chức năng
00	Không đổi
01	Cho phép nạp giá trị mới
10	Dịch trái 1 bit
11	Dịch phải 1 bit

Bảng 1: Bảng sự thật Shift Register

Bảng sự thật Register

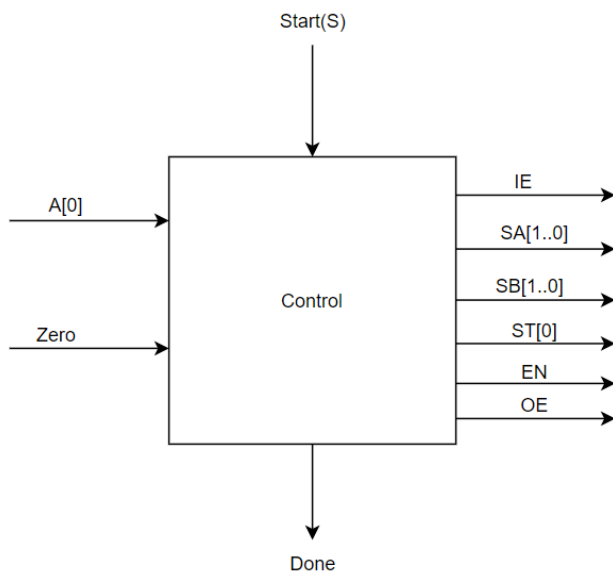
RS	WS	O (ngõ ra)
0	X	Z
1	0	Q
1	1	I

Bảng 2: Bảng sự thật Register

- Giải thích cách thức hoạt động
- + B1: Cho phép nhập các giá trị A, B. Gán Tmp = 0: IE = 1 cho phép nhập, SA = 01 cho phép ghi, SB = 01 cho phép ghi, ST = 1 nạp 0 vào thanh ghi, EN = 0 không cho phép bật counter, OE = 0 không cho phép xuất kết quả
- + B2: IE = 0 không cho phép nhập, SA = 00, SB = 00, ST = 0 để giữ nguyên giá trị, EN = 1 bật counter, OE = 0 không cho phép xuất kết quả
- + B3: SA = 00, SB = 00 giữ nguyên giá trị, ST = 1 cho phép ghi, dựa vào A[0] = 0 hay 1, nếu bằng 0 thì Tmp = Tmp, nếu bằng 1 thì Tmp = Tmp + B. Tắt counter EN = 0, không cho phép xuất kết quả OE = 0
- + B4: IE = 0 không cho phép ghi, SA = 11 để dịch phải, SB = 10 để dịch trái, ST = 0 giữ nguyên giá trị, EN = 0 không bật counter, OE = 0 không cho phép xuất kết quả. Rồi kiểm tra điều kiện đã lặp 32 lần chưa của counter, nếu đã lặp đủ thì qua B5, nếu chưa thì quay lại B2
- + B5: IE = X giá trị tùy định, SA = 00, SB = 00, ST = 0 để giữ nguyên giá trị, EN = không bật counter, OE = 1 cho phép xuất kết quả.

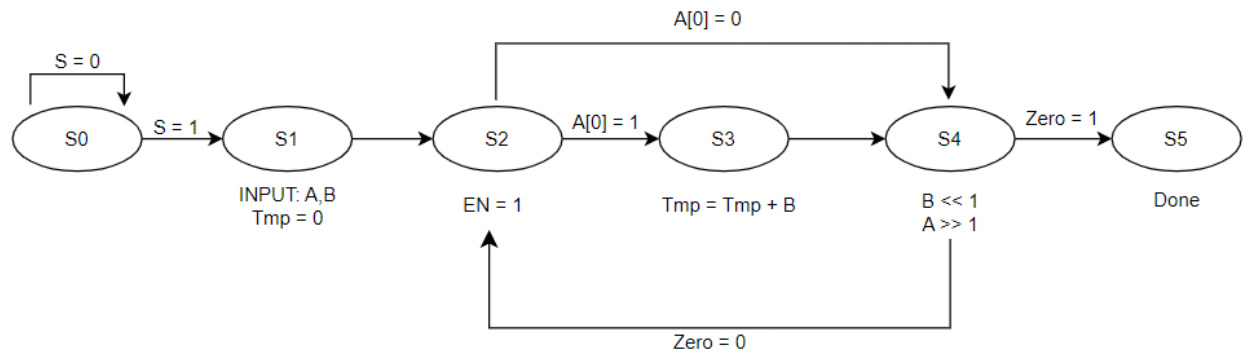
2.1.3. Control

- Tương ứng với output của Datapath sẽ là Input của Control là Zero và A[0] và tín hiệu bắt đầu thực hiện S
- Output của khối control tương ứng với Input của Datapath là IE, SA[1..0], SB[1..0], ST[0], EN, OE và tín hiệu báo Done khi hoàn thành phép nhân



Hình 3: Khối control phép nhân

2.1.3.1. Lưu đồ chuyển trạng thái



Hình 4: Lưu đồ chuyển trạng thái phép nhân

Control word

	Q2Q1Q0	S,A,Z							
		000	001	010	011	100	101	110	111
s0	000	000	000	000	000	001	001	001	001
s1	001	010	010	010	010	010	010	010	010
s2	010	100	100	011	011	100	100	011	011
s3	011	100	100	100	100	100	100	100	100
s4	100	010	101	010	101	010	101	010	101
s5	101	000	000	000	000	000	000	000	000

Bảng 3: Bảng control word phép nhân

Phương trình ngõ vào D Flip Flop

$$D2 = A'Q2'Q1 + Q2'Q1Q0 + ZQ2Q1'Q0'$$

$$D1 = Q2'Q1'Q0 + Z'Q2Q1'Q0' + AQ2'Q1Q0'$$

$$D0 = ZQ2Q1'Q0' + AQ2'Q1Q0' + SQ2'Q1'Q0'$$

Bảng cho các ngõ ra

	Q2Q1Q0	Step	IE	SA[1..0]	SB[1..0]	ST[0]	EN	OE
s0	000	0	X	00	00	0	0	0
s1	001	1	1	01	01	1	0	0
s2	010	2	0	00	00	0	1	0
s3	011	3	0	00	00	1	0	0
s4	100	4	0	11	10	0	0	0
s5	101	5	X	00	00	0	0	1

Bảng 4: Bảng ngõ ra khỏi control phép nhân

Phương trình ngõ ra:

$$IE = Q2'Q1'$$

$$SA[1] = Q2Q1'Q0'$$

$$SA[0] = Q2'Q1'Q0 + Q2Q1'Q0'$$

$$SB[1] = Q2Q1'Q0'$$

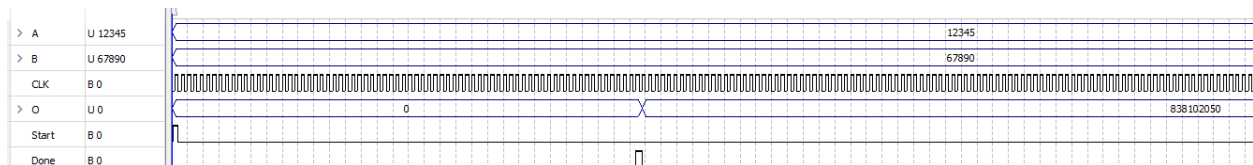
$$SB[0] = Q2'Q1'Q0$$

$$ST[0] = Q2'Q0$$

$$EN = Q2'Q1Q0'$$

$$OE = Done = Q2Q1'Q0$$

2.1.4. Kết quả mô phỏng

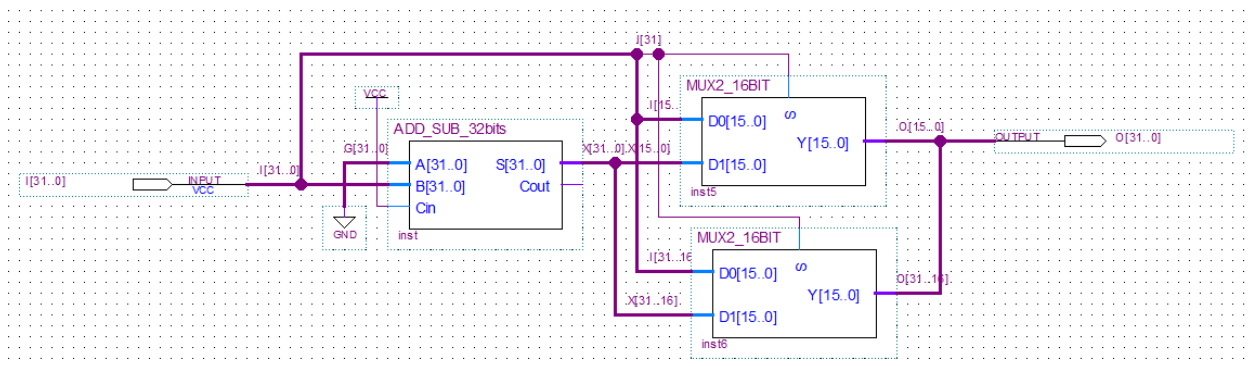


Hình 5: kết quả mô phỏng phép nhân

2.1.5. Cải tiến phép nhân có dấu từ phép nhân không dấu

- Sử dụng toàn bộ như datapath, controller đã dùng ở trên
- Thay đổi:

Chuyển input thành số dương (nếu âm)

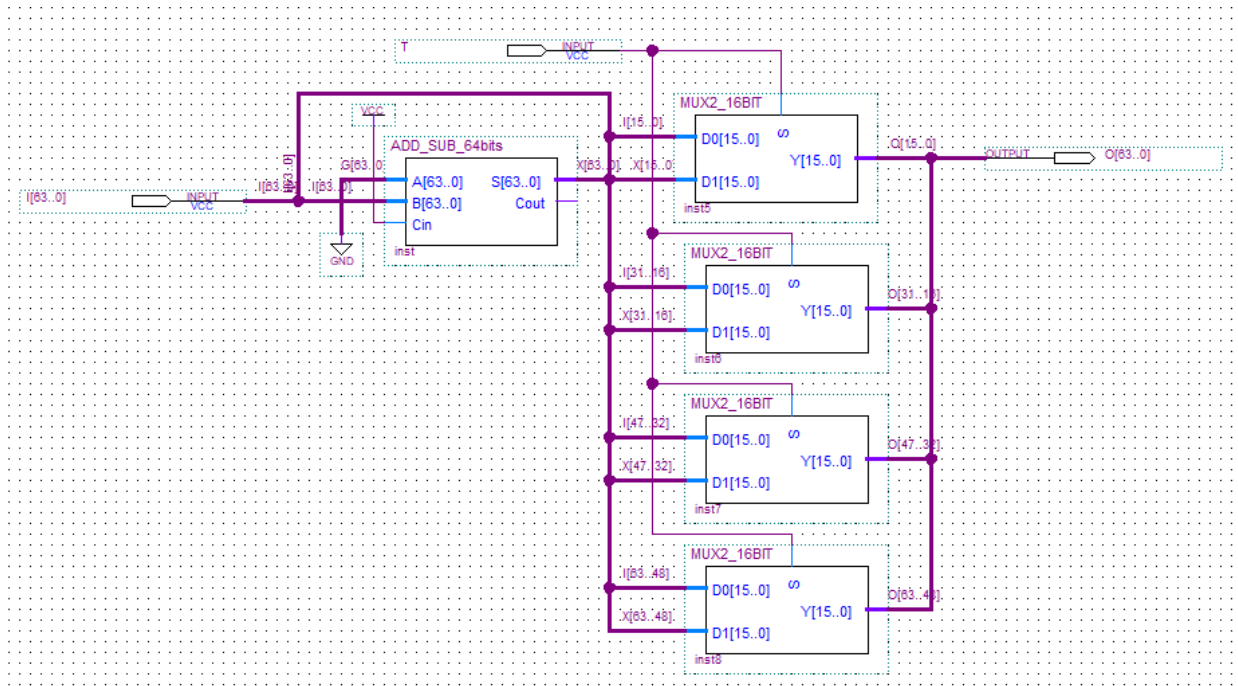


Hình 6: Mạch chuyển nhân không dấu sang nhân có dấu

$M = \text{XOR 2 bit dấu của input}$

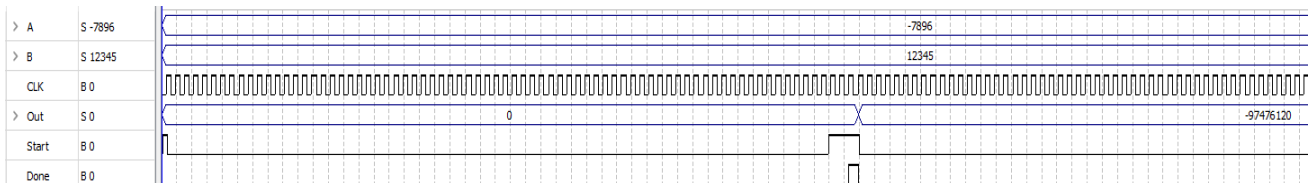
Nếu $M = 0$, xuất output ra ngoài

$M = 1$, chuyển output thành số âm



Hình 7: Mạch phép nhân có dấu

Mô phỏng:

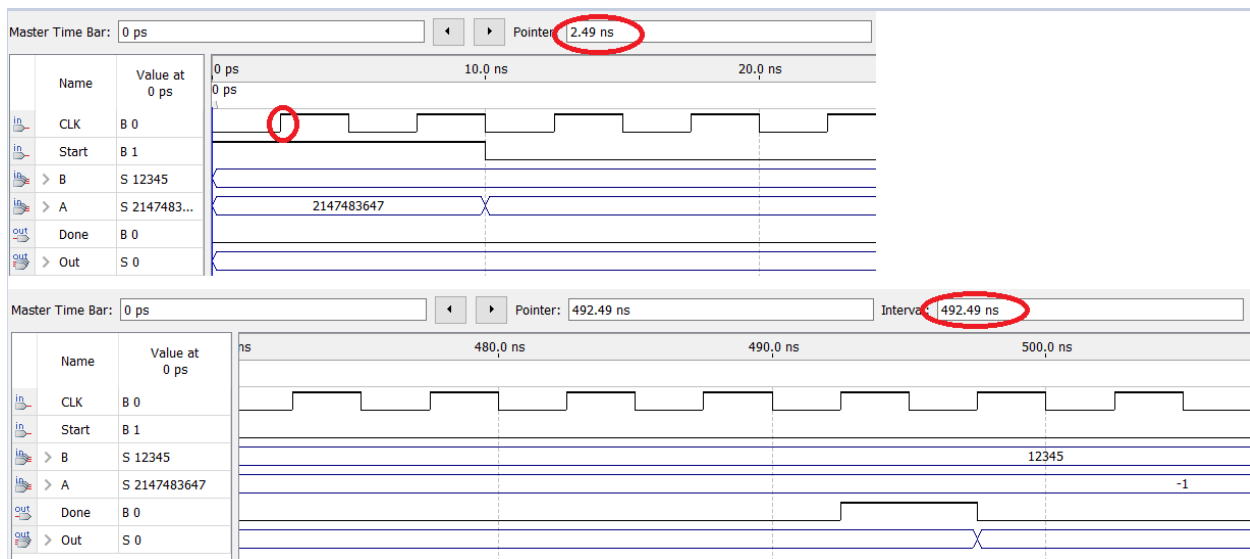


Hình 8: Mô phỏng kết quả phép nhân có dấu

2.1.6. Đánh giá kết quả phép nhân

- Trong trường hợp xấu nhất khi $A = 01...111$, B tùy chọn (2147483647 trong hệ 10, vì bit đầu tiên là bit dấu nên chỉ cho 31 bit sau là 1) thì sẽ tốn:

$$1\text{CK (nhập)} + 32 * 3 + 1\text{CK (xuất kết quả)} = 98\text{ CK (dựa theo sơ đồ chuyển trạng thái)}$$



Hình 9: Đánh giá kết quả phép nhân - 1

Chu kỳ mô phỏng là 5ns

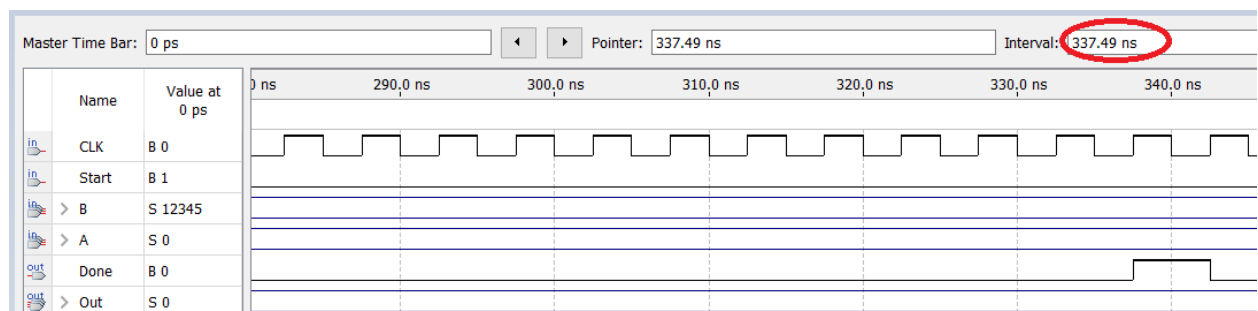
Ngay tại lần tích cực đầu tiên của CLK (Start = 1) thì thời gian là 2.5ns, sau khi đến 492.5ns thì ngõ ra Done = 1

Vậy thời gian thực hiện khi mô phỏng là: $492.5 - 2.5 = 490\text{ns}$

⇒ Thời gian thực hiện là $490 / 5 = 98 =$ lý thuyết đã tính

- Trong trường hợp tốt nhất $A = 0...0$ (32 bit 0), B tùy chọn thì sẽ tốn:

$$1\text{CK}(\text{nhập}) + 31 * 2 + 1\text{CK}(\text{xuất kết quả}) = 66 \text{ CK}$$



Hình 10: Đánh giá kết quả phép nhân - 2

Chu kỳ mô phỏng là 5ns

Lần tích cực đầu tiên của CLK là 2.5ns (Start = 1), khi đến 337.5ns thì Done = 1

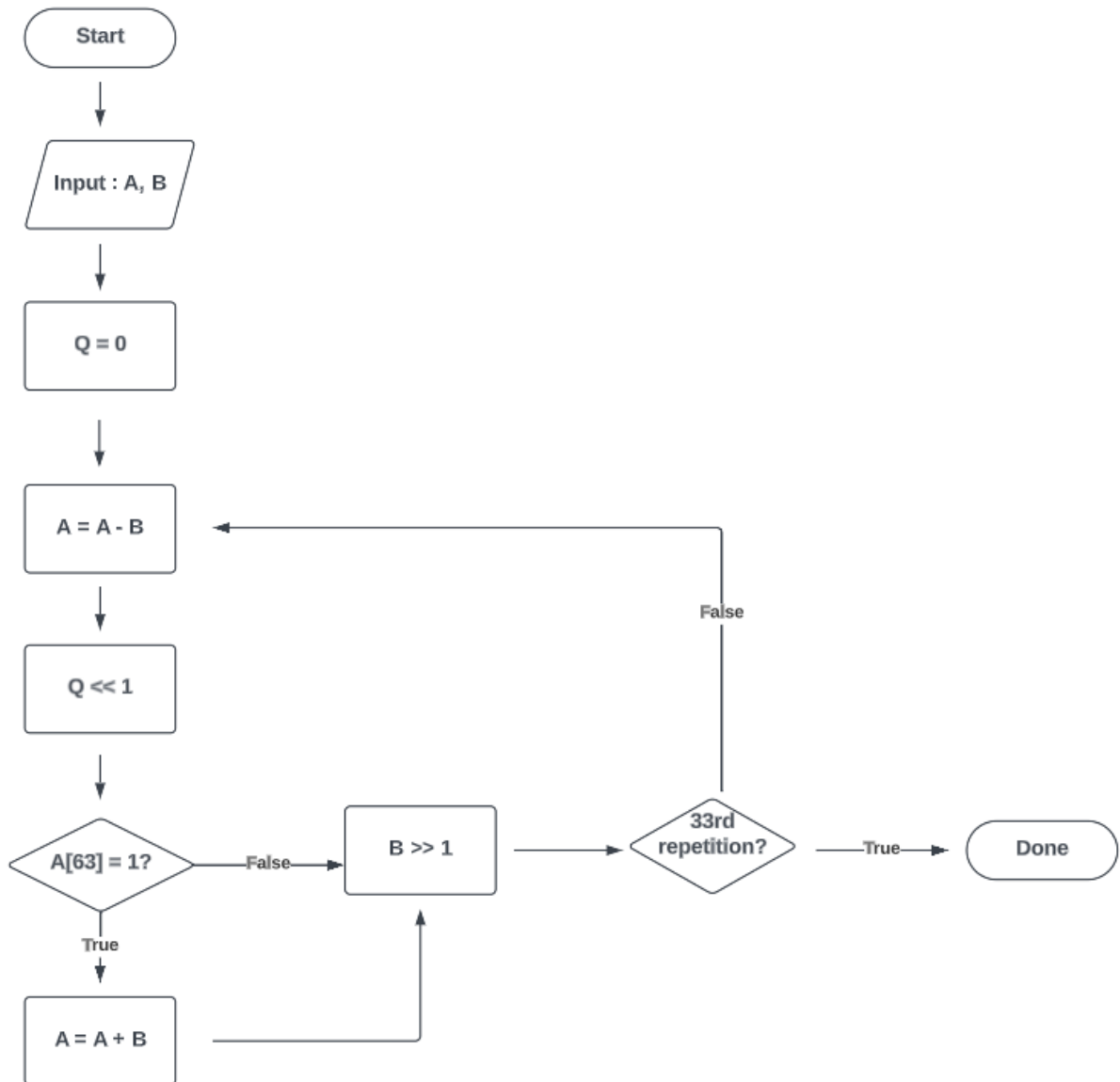
Vậy thời gian thực hiện khi mô phỏng là: $337.5 - 2.5 = 335\text{ns}$

⇒ Thời gian thực hiện là $335 / 5 = 67 \text{ CK} =$ lý thuyết đã tính + 1

- Vậy thì với cách thực hiện phép nhân như đã trình bày thì số chu kỳ cần thiết để thực hiện (CK) là $67 \leq CK \leq 98$

2.2. Phép chia 32 bits số nguyên không dấu

2.2.1. Lưu đồ giải thuật



Hình 11: Lưu đồ giải thuật phép chia

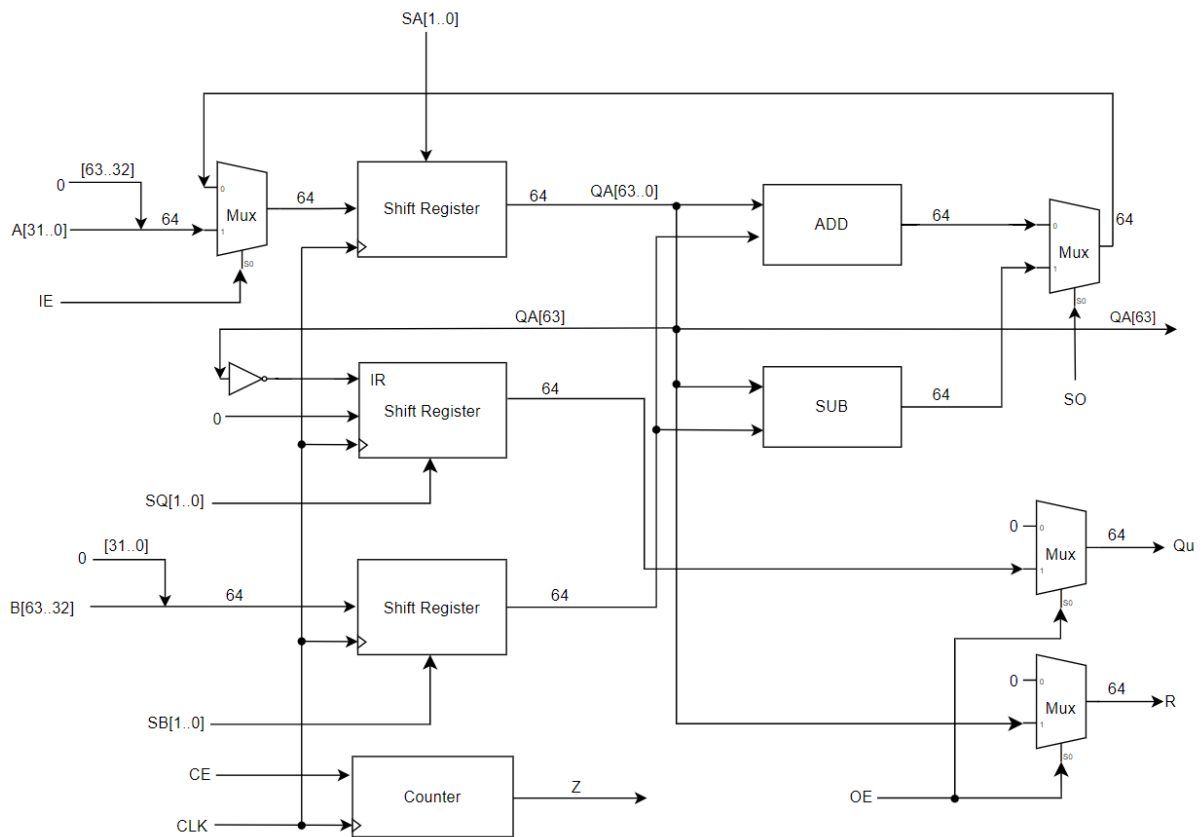
2.2.2. Datapath

Dựa vào lưu đồ giải thuật, đầu tiên ta cần xác định được những thành phần cần thiết để tạo nên datapath

- Thanh ghi dịch để lưu các giá trị A (số bị chia / số dư), B(số chia), Q (kết quả tạm thời) và thực hiện dịch bit
- Bộ counter thực hiện việc đếm cho vòng lặp 32 lần

- Bộ cộng, trừ để lấy kết quả tạm thời
- Các mux2to1 64 bits để lựa chọn đầu ra theo điều kiện

Hình vẽ dưới đây mô tả đường dữ liệu cho phép tính nhân



Hình 12: Datapath phép chia

- Input:
 - + A: 64 bits, lưu số bị chia
 - + SA[1..0]: điều khiển thanh ghi lưu A
 - + B: 64bits, lưu số chia
 - + SB[1..0]: điều khiển thanh ghi lưu B
 - + Q: 64bits, lưu kết quả khi tính toán
 - + SO: lựa chọn phép tính tương ứng với điều kiện theo QA[63]
 - + SQ[1..0]: điều khiển thanh ghi lưu Q
 - + CE: điều khiển hoạt động của counter
 - + OE: tín hiệu cho phép xuất kết quả
- Output:
 - + Z: tín hiệu của counter khi lặp đủ 33 lần
 - + QA[63]: tín hiệu điều kiện lựa chọn phép tính
 - + output: gồm Qu 64 bits là kết quả, R 64 bits là phần dư
- Giải thích cách thức hoạt động:

+ B1: Cho phép nhập giá trị $IE = 1$, $SA = 01$, $SB = 01$ cho phép ghi giá trị A,B nhập vào, $SQ = 01$ cho phép ghi 0 vào thanh ghi, $SO = X$ lựa chọn phép tính là tùy định, $CE = 0$ không bật counter, $OE = 0$ không cho phép xuất

+ B2: $IE = 0$ không cho phép nhập giá trị, $SA = 01$ cho phép nạp giá trị mới, $SB = 00$, $SQ = 00$ để giữ nguyên giá trị, $SO = 1$ để lựa chọn phép trừ, $CE = 1$ bật counter, $OE = 0$ không cho phép xuất. Vì phép trừ chỉ tốn 1 chu kỳ thực hiện nên có thể bật counter ở bước này. Sau B2 thì $A = A - B$.

+ B3: $IE = 0$ không cho phép ghi giá trị, $SA = 00$, $SB = 00$ giữ nguyên giá trị, $SQ = 10$ để dịch trái thanh ghi chứa kết quả, SO tùy định, $CE = 0$, $OE = 0$. Kết quả dịch trái của Q phụ thuộc vào kết quả của $A = A - B$ của B2. Nếu kết quả âm thì Q dịch trái với bit trọng số thấp nhất là 0, nếu kết quả là dương thì bit có trọng số thấp nhất là 1

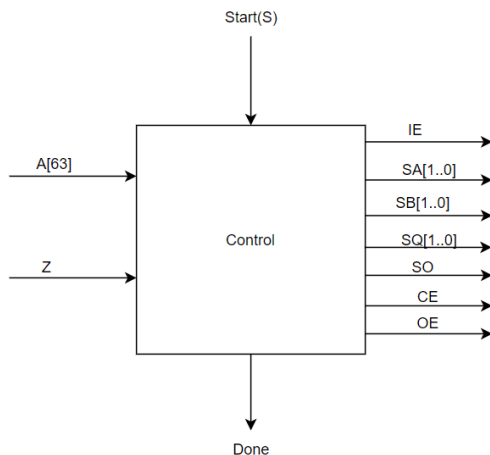
+ B4: Nếu $QA[63] = 1$, có nghĩa là kết quả của phép trừ là số âm thì $A = A + B$, $SA = 01$ để nạp lại giá trị cho A. Nếu $QA[63] = 0$ thì bỏ qua bước này. $IE = 0$, $CE = 0$, $OE = 0$, $SB = 00$, $SQ = 00$ và $SO = 0$ để lựa chọn phép cộng.

+ B5: Dịch phải thanh ghi chứa B, $SB = 11$. $SA = 00$, $IE = 0$, $SQ = 00$, $SO = X$, $OE = 0$, $CE = 0$. Kiểm tra đã lặp đủ 33 lần chưa, nếu đã đủ qua B6, nếu chưa quay lại B2

+ B6: Xuất kết quả cuối cùng: $OE = 1$, $IE = 0$, $SA = 00$, $SB = 00$, $SQ = 00$, $SO = X$, $CE = 0$

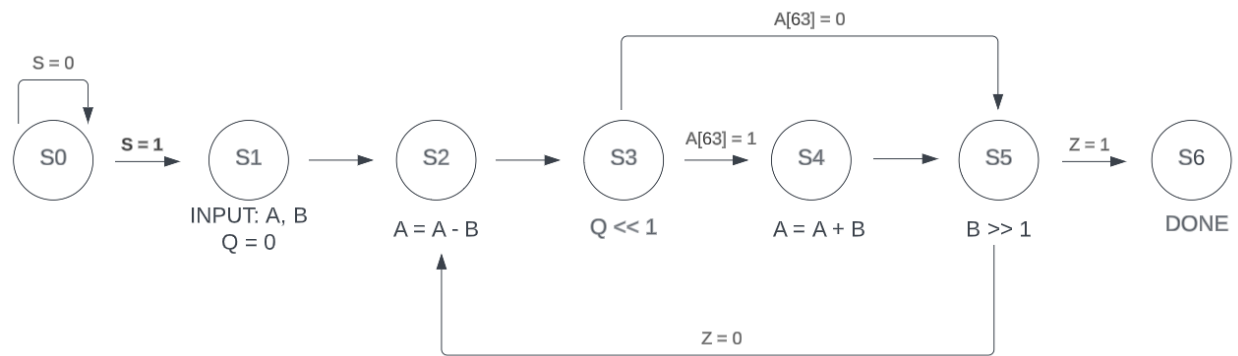
2.2.3. Control

- Tương ứng với output của Datapath sẽ là Input của Control là Z và A[63] và tín hiệu bắt đầu thực hiện S
- Output của khối control tương ứng với Input của Datapath: IE , $SA[1..0]$, $SB[1..0]$, $SQ[1..0]$, SO , CE , OE và tín hiệu báo Done khi hoàn thành phép chia



Hình 13: Khối control phép chia

2.1.3.1. Lưu đồ chuyển trạng thái



Hình 14: Lưu đồ chuyển trạng thái phép chia

Bảng control word

STEP	IE	SA	SB	SQ	SO	OE	CE
0	0	00	00	00	X	0	0
1	1	01	01	01	X	0	0
2	0	01	00	00	1	0	1
3	0	00	00	10	X	0	0
4	0	01	00	00	0	0	0
5	0	00	11	00	X	0	0
6	0	00	00	00	X	1	0

Bảng 5: Bảng control word phép chia

A: Remainder/Dividend (số dư/số bị chia)

Q: Quotient (thương số)

B: Divisor (số chia)

QA[63]: bit trọng số cao nhất của A

Bảng ngõ vào Flip Flop D

TTHT	TTKT/NGÕ VÀO FF Q2Q1Q0								
	SCZ								
	000	001	010	011	100	101	110	111	
000	000	000	000	000	001	001	001	001	S0
001	010	010	010	010	010	010	010	010	S1
010	011	011	011	011	011	011	011	011	S2
011	101	101	100	100	101	101	100	100	S3
100	101	101	101	101	101	101	101	101	S4
101	010	110	010	110	010	110	010	110	S5

110	000	000	000	000	000	000	000	000	S6
111	000	000	000	000	000	000	000	000	S7

Bảng 6: Bảng ngõ vào DFF phép chia

Phương trình ngõ vào D:

$$D2 = Q2'Q1Q0 + Q2Q1'Q0' + Q2Q1'Z$$

$$D1 = Q1'Q0 + Q2'Q1Q0'$$

$$D0 = Q2'Q0'S + Q2'Q1Q0' + Q2'Q1C' + Q2Q1'Q0'$$

Bảng ngõ ra khối control

TTHT	IE	SA	SB	SQ	SO	OE	CE	Trạng thái
000	0	00	00	00	X	0	0	S0
001	1	01	01	01	X	0	0	S1
010	0	01	00	00	1	0	1	S2
011	0	00	00	10	X	0	0	S3
100	0	01	00	00	0	0	0	S4
101	0	00	11	00	X	0	0	S5
110	0	00	00	00	X	1	0	S6
111	0	00	00	00	X	0	0	S7

Bảng 7: Bảng ngõ ra khối control phép chia

$$IE = Q2'Q1'Q0$$

$$SA[1] = 0$$

$$SA[0] = Q2'Q1'Q0 + Q2'Q1Q0' + Q2Q1'Q0'$$

$$SB[1] = Q1Q1'Q0$$

$$SB[0] = Q1'Q0$$

$$SQ[1] = Q2'Q1Q0$$

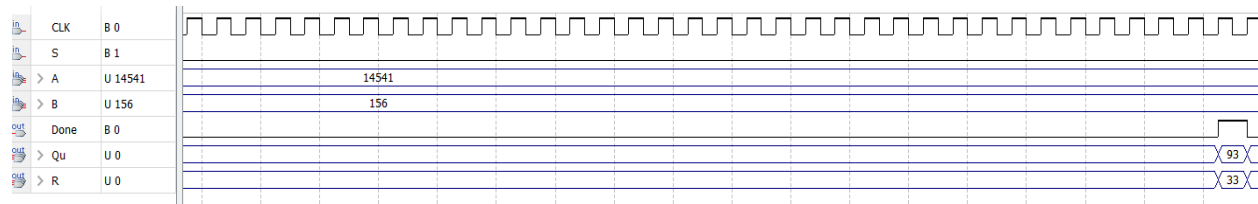
$$SQ[0] = Q2'Q1'Q0$$

$$SO = Q2'$$

$$OE = Done = Q2Q1Q0'$$

$$CE = Q2'Q1Q0'$$

2.2.4. Kết quả mô phỏng



Hình 15: kết quả mô phỏng phép chia

2.2.5. Cải tiến phép chia có dấu từ phép chia không dấu

Chuyển input thành số dương (nếu âm)

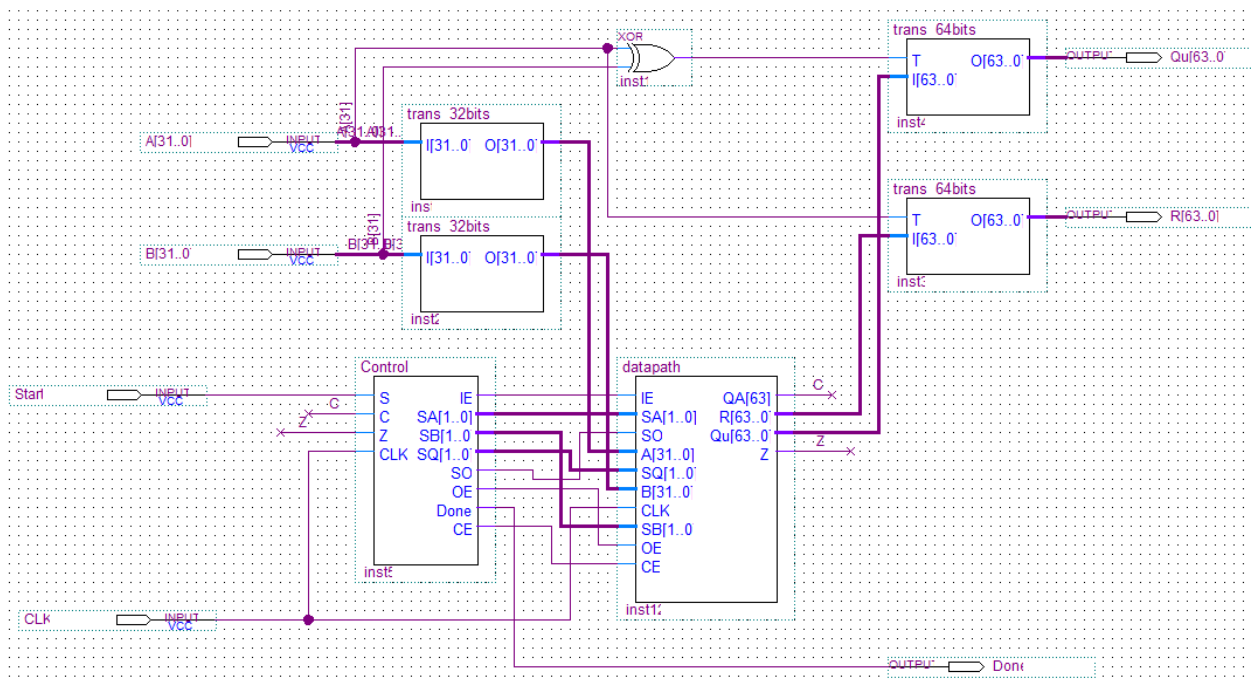
$M1 = \text{XOR}$ 2 bit dấu của input dùng cho Qu

Nếu $M1 = 1$ chuyển Qu thành số âm

$M = 0$ chuyển Qu thành số dương

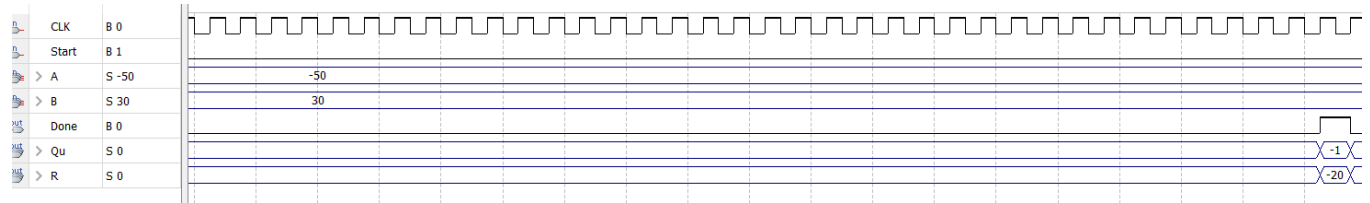
$M2 = \text{bit}$ dấu của A

$M2 = 1 \rightarrow R$ thành âm, ngược lại R thành dương



Hình 16: Mạch phép chia có dấu

Mô phỏng:



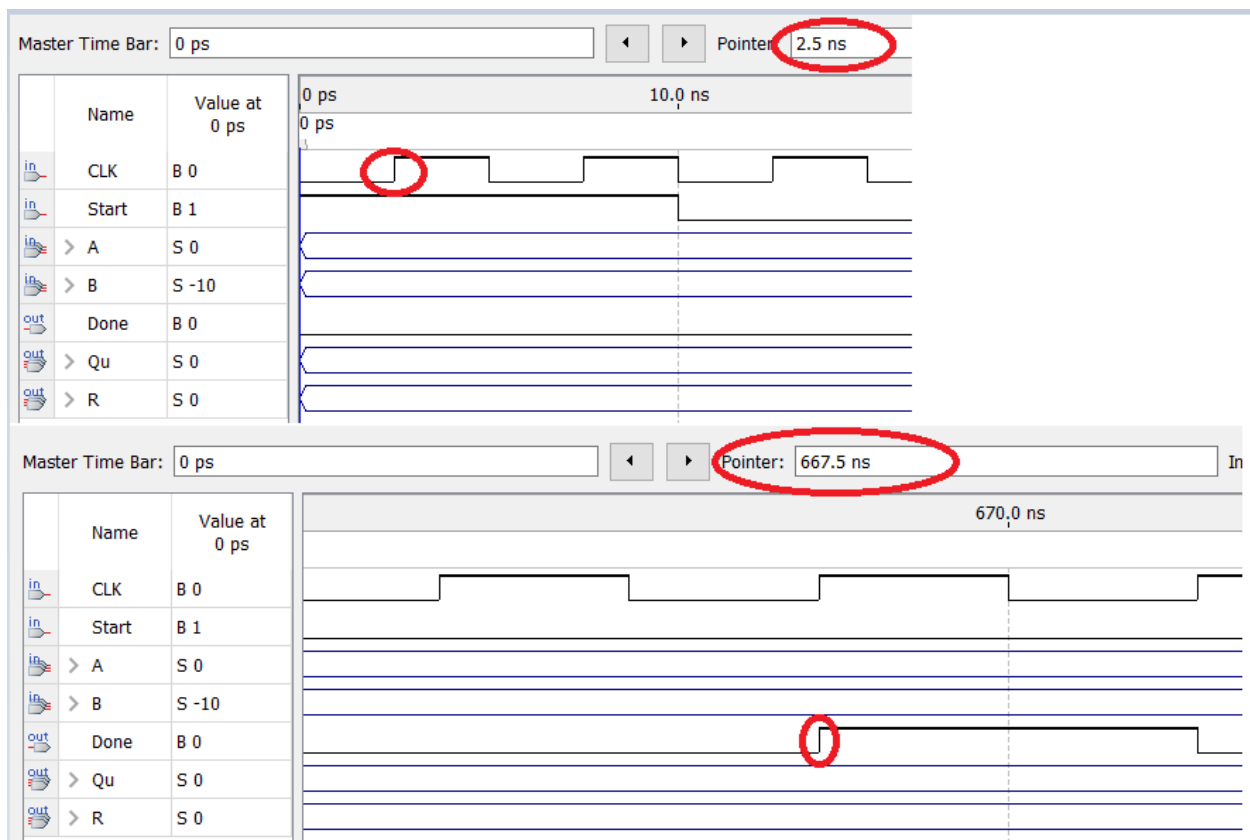
Hình 17: Mô phỏng kết quả phép chia có dấu

2.2.6. Đánh giá kết quả phép chia

- Trong trường hợp xấu nhất khi $A = 0...00$ (0 trong hệ 10), $B \neq 0$ thì sẽ tốn:

$1CK$ (nhập) + $32 * 4 + 2 + 1CK$ (xuất kết quả) = $132 CK$ (dựa theo sơ đồ chuyển trạng thái)

Giải thích vì sao $32 * 4 + 2$: vì 32 chu kỳ đầu tiên thì A luôn là 0 là B là số khác 0 nên là $32 * 4$, còn chu kỳ thứ 33 vì B đã dịch hết sang phải 32 lần, mà số đã nhập là 32 bit nên B là 0, $A - B = 0 - 0 = 0$ nên chỉ tốn 2 ck (s2 và s5) ở lần thứ 33



Hình 18: Đánh giá kết quả phép chia - 1

