

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA



## ĐỒ ÁN THIẾT KẾ LUẬN LÝ (CO3091)

---

Mạch chia - trừ 2 số nguyên dương có nhớ kết quả

---

GVHD: Đoàn Minh Vững  
Hội đồng báo cáo:  
SV thực hiện: Trịnh Cao Thắng – 2014550  
Nghiêm Lương Sơn – 2014373

**Phân chia công việc.**

Thành viên	Công việc
Trịnh Cao Thắng	* Xây dựng và mô phỏng mạch tính toán phép chia và phép trừ. * Viết báo cáo.
Nghiêm Lương Sơn	* Xây dựng các mạch cho tín hiệu Input và Output. * Viết báo cáo.

## Contents

<b>1</b>	<b>Đề tài</b>	<b>3</b>
1.1	Tổng quan đề tài . . . . .	3
1.2	Phương thức thực hiện . . . . .	3
<b>2</b>	<b>Các quy ước cơ bản</b>	<b>3</b>
<b>3</b>	<b>Cơ sở Lý thuyết</b>	<b>3</b>
3.1	Mạch trừ . . . . .	3
3.2	Mạch chia . . . . .	5
<b>4</b>	<b>Các linh kiện dùng trong mạch</b>	<b>6</b>
4.1	Khối tín hiệu Input và Output . . . . .	6
4.1.1	Vi điều khiển 8051 . . . . .	6
4.1.2	IC 7490 . . . . .	7
4.1.3	IC chốt dữ liệu 74HC573 . . . . .	10
4.2	Khối xử lý . . . . .	12
4.2.1	IC 74LS83 . . . . .	12
4.2.2	IC 74LS386 . . . . .	14
<b>5</b>	<b>Source code dùng trong Đề tài</b>	<b>17</b>
<b>6</b>	<b>Hiện thực các mạch thành phần</b>	<b>18</b>
6.1	Hiện thực mạch chia . . . . .	18
6.1.1	Khối S . . . . .	18
6.1.2	Khối $SUB_i$ . . . . .	19
6.1.3	Khối Remainder Modify . . . . .	19
6.1.4	Kết hợp thành mạch chia hoàn chỉnh . . . . .	20
6.2	Hiện thực mạch trừ . . . . .	20
6.3	Hiện thực mạch lưu . . . . .	21
<b>7</b>	<b>Tổng hợp các mạch thành phần</b>	<b>22</b>
<b>8</b>	<b>Kiểm thử chức năng</b>	<b>23</b>
8.1	Ví dụ 1 . . . . .	23
8.2	Ví dụ 2 . . . . .	23
8.3	Ví dụ 3 . . . . .	24
<b>9</b>	<b>Đánh giá</b>	<b>25</b>
9.1	Ưu điểm . . . . .	25
9.2	Nhược điểm . . . . .	25
9.3	Giải pháp . . . . .	25
<b>10</b>	<b>Phân công công việc và kế hoạch thực hiện đồ án</b>	<b>26</b>
<b>11</b>	<b>Tài liệu tham khảo</b>	<b>26</b>

# 1 Đề tài

## 1.1 Tổng quan đề tài

- Đề tài: Thực hiện mạch trừ và mạch chia một số có 2 chữ số cho một số có 1 chữ số có nhớ kết quả.
- Kiến thức chủ yếu của đề tài dựa vào logic, điện tử và kỹ thuật số cơ bản.
- Linh kiện được dùng trong đề tài xoay quanh các cổng logic như AND, OR, NOT, XOR ...; các loại IC với nhiều chức năng; các bộ vi xử lý, vi điều khiển.
- Kiểm tra kiến thức cơ bản trong môn học thiết kế luận lý, điện – điện tử đã học từ đó vận dụng để thiết kế được các mạch đơn giản, mô phỏng để kiểm tra tính đúng đắn, tính logic của mạch đó.

## 1.2 Phương thức thực hiện

- Tìm hiểu các linh kiện sử dụng cho mạch.
- Tìm hiểu công nghệ, phần mềm sử dụng để mô phỏng.
- Vận dụng các kiến thức để kết nối tạo ra sản phẩm hoàn thiện.
- Làm gọn dây, tối ưu hóa linh kiện sử dụng và mô phỏng mạch để kiểm tra.

# 2 Các quy ước cơ bản

- Đối với mạch trừ: số bị trừ A là số có 2 chữ số nên A có 8 bits, số trừ B là số có 1 chữ số nên B có 4 bits và hiệu C sẽ có 8 bits.
- Đối với mạch chia: số bị chia A là số có 2 chữ số nên A có 8 bits, số chia B là số có 1 chữ số nên B có 4 bits, thương số C sẽ có 8 bits và số dư D sẽ có 4 bits.
- Vì số bị trừ và số số bị chia là số có 2 chữ số ( $10 \leq x \leq 99$ ) nên nhóm chúng em mặc định bit thứ 8 (MSB bit) là bit 0.

# 3 Cơ sở Lý thuyết

## 3.1 Mạch trừ

### Ý tưởng hiện thực

- Chuyển số trừ B sang hệ bù 2 bằng cách đảo tất cả các bits của B và sau đó cộng thêm bit 1.
- Thực hiện phép cộng số bị trừ A cho số bù 2 của số trừ B thông qua mạch cộng toàn phần Full Adder.

**Mạch cộng Full Adder:** Là mạch cộng 2 bit ở cùng vị trí trong hai số nhị phân nhiều bit, nói cách khác, đây là mạch cộng 2 bit, giả sử thứ n, và bit nhớ có được từ phép cộng hai bit thứ n-1 của hai số nhị phân đó.

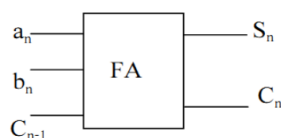


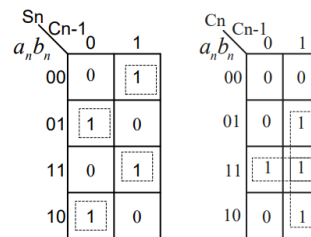
Figure 1: Sơ đồ mô phỏng mạch cộng toàn phần Full Adder

Trong đó:

- $a_n$  là số bị trừ
- $b_n$  là số trừ
- $C_{n-1}$  là số nhớ của lần cộng trước đó
- $C_n$  là số nhớ của lần cộng hiện tại
- $S_n$  là tổng hiện tại

$a_n$	$b_n$	$C_{n-1}$	$S_n$	$C_n$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**Figure 2:** Bảng chân trị của mạch Full Adder

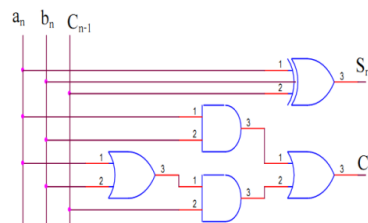


**Figure 3:** Bìa Karnaugh cho các hàm output của mạch Full Adder

Từ bìa Karnaugh, ta có thể biểu diễn phương trình trạng thái rút gọn cho hai hàm  $C(n)$  và  $S(n)$  như sau :

$$S(n) = \overline{a_n} \overline{b_n} C_{n-1} + \overline{a_n} b_n \overline{C_{n-1}} + a_n b_n C_{n-1} + a_n \overline{b_n} \overline{C_{n-1}} = a_n \oplus b_n \oplus C_{n-1}$$

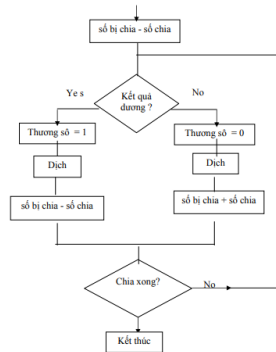
$$C(n) = b_n C_{n-1} + a_n C_{n-1} + a_n b_n = a_n b_n + C_{n-1} (a_n + b_n)$$



**Figure 4:** Sơ đồ mạch cộng toàn phần Full Adder

### 3.2 Mạch chia

Nguyên tắc của phép chia số nhị phân là thực hiện phép so sánh một phần của số bị chia (số bit đầu tiên bằng với số bit của số chia) với số chia, nếu số bị chia lớn hơn số chia thì thương số = 1, thực hiện phép trừ, ngược lại thì thương số = 0, sau đó dịch trái phần còn lại của số bị chia 1 bit (hoặc dịch phải số chia 1 bit) rồi tiếp tục thực hiện bài toán so sánh giống như trên. Công việc được lặp lại cho đến khi chấm dứt.



**Figure 5:** Flowchart cho thuật toán thực hiện phép chia

Số chia lớn hơn số bị chia (nhánh bên phải):

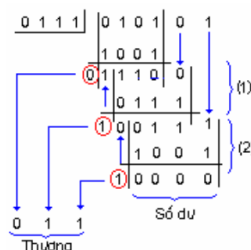
- Lưu ý là dịch số chia về bên phải 1 bit tương đương với chia số đó cho 2.
- Nhánh bên phải của sơ đồ trên gồm 2 bài toán:
  - Cộng số bị chia với số chia.
  - Trừ số bị chia cho 1/2 số chia (trừ bị chia cho số chia đã dịch phải).
- Hai bước này có thể gom lại thành một bước duy nhất như sau: Cộng số bị chia với số chia đã dịch phải.

Số chia nhỏ hơn số bị chia (nhánh bên trái): Sau khi lấy kết quả = 1, lệnh kế tiếp thực hiện là trừ số chia đã dịch phải.

Các bước thực hiện bài toán như sau:

Bước 1 : Số chia (SC) lớn hơn số bị chia (SBC) ( $SBC - SC < 0$ ), thương số là 0, dịch phải số chia 1 bit (thực tế ta mang thêm 1 bit của số bị chia xuống), thực hiện bài toán cộng số chia và số bị chia.

Bước 2 : Số chia nhỏ hơn số bị chia ( $SBC - SC > 0$ ), thương số là 1, dịch phải số chia 1 bit, thực hiện bài toán trừ (cộng số bù 2) số bị chia cho số chia.



**Figure 6:** Thực hiện chia số  $21_{10}$  ( $010101_2$ ) cho số  $7_{10}$  ( $0111_2$ )

Ghi chú:

- Số 1 trên mũi tên chỉ rằng kết quả phép toán trừ là số âm, bước kế tiếp là dời và cộng số chia.
- Số 0 trên mũi tên chỉ rằng kết quả phép toán trừ là số dương, bước kế tiếp là dời và trừ số chia (cộng số bù 2)
- Thương số có được từ các số tràn mà trên phép tính ta ghi trong vòng tròn.
- Kết quả: thương là  $011_2 = 3$  và số dư là  $0000_2 = 0$

## 4 Các linh kiện dùng trong mạch

### 4.1 Khối tín hiệu Input và Output

#### 4.1.1 Vi điều khiển 8051

Vi điều khiển 8051 được Intel thiết kế vào năm 1981. Đây là bộ vi điều khiển 8 bit, được xây dựng với 40 chân DIP (gối nội tuyến kép), 4kb bộ nhớ ROM và 128 byte bộ nhớ RAM, 2 bộ định thời 16 bit. Nó bao gồm bốn cổng 8 bit song song, có thể lập trình cũng như định địa chỉ theo yêu cầu. Một bộ dao động tinh thể trên chip được tích hợp trong bộ vi điều khiển có tần số tinh thể là 12 MHz.

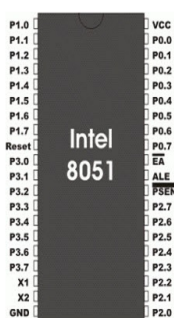


Figure 7: Sơ đồ chân của vi điều khiển 8051



Figure 8: Hình ảnh thực tế ngoài thị trường của vi điều khiển 8051

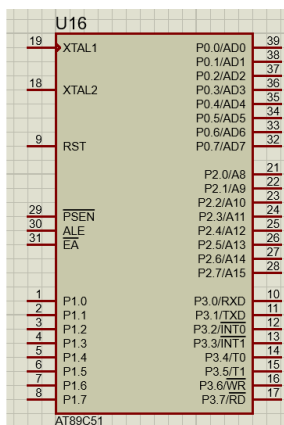


Figure 9: Hình ảnh vi điều khiển 8051 trong phần mềm Proteus

### Lịch sử của vi điều khiển 8051

- Nếu chúng ta quay ngược lại lịch sử, bộ vi điều khiển 8051 được phát minh lần đầu tiên vào năm 1980 bởi gã khổng lồ vi xử lý Intel và dần dần nó đã được chấp nhận trên toàn thế giới và càng ngày tầm quan trọng của bộ vi điều khiển 8051 càng tăng lên. Khi được phát minh bởi Intel, nó được phát triển bằng công nghệ NMOS, nhưng không hiệu quả lắm.
- Để tăng hiệu quả và năng suất, Intel đã sửa lại nó bằng cách áp dụng công nghệ CMOS và một phiên bản mới đã ra đời với chữ cái C trong tên để nó có thể đáp ứng nhu cầu và có thể hoạt động tối đa. Phiên bản mới của vi điều khiển 8051 có hai bus và trong đó một bus dành cho chương trình và bus khác dành cho dữ liệu để nó có thể hoạt động tốt.
- Vi điều khiển 8051 là dòng vi điều khiển 8 bit và được sử dụng trên toàn thế giới. Hệ thống trên chip là từ đồng nghĩa của vi điều khiển 8051 và các thành phần mà nó có bao gồm 128 byte RAM, bốn cổng trên một chip, 2 bộ định thời, 1 cổng nối tiếp và 4Kbyte ROM.
- Vì nó là bộ xử lý 8 bit nên CPU có thể hoạt động rất hiệu quả và nhanh chóng nếu dữ liệu khoảng 8 bit tại một thời điểm, nếu dữ liệu nhiều hơn nó phải được phân mảnh thành nhiều CPU khác nhau. Thực tế, trong thời đại ngày nay, hầu hết của các nhà sản xuất thích đi kèm với 4Kbyte ROM.

Ứng dụng của vi điều khiển 8051:

- Quản lý năng lượng: Vi điều khiển 8051 được trang bị hệ thống đo lường hiệu quả và nó giúp vi điều khiển tiết kiệm năng lượng ở mức độ lớn.
- Màn hình cảm ứng: những vi điều khiển hiện đại có tính năng màn hình cảm ứng và bộ vi điều khiển 8051 cũng đi kèm với tính năng màn hình cảm ứng. Do đó nó có một ứng dụng rộng rãi trong điện thoại di động, máy nghe nhạc và game.
- Lĩnh vực ô tô: vi điều khiển 8051 có một ứng dụng rộng rãi trong lĩnh vực ô tô và đặc biệt là trong quản lý xe hybrid. Ngoài ra hệ thống kiểm soát hành trình và chống phanh là lĩnh vực khác mà nó có công dụng rất lớn.

#### 4.1.2 IC 7490

IC 7490 là bộ đếm thập phân không có chức năng reset tự động. IC hoạt động dựa trên một Flip Flop loại T, chúng có thể tạo ra bộ đếm có  $2n$  số nhị phân.

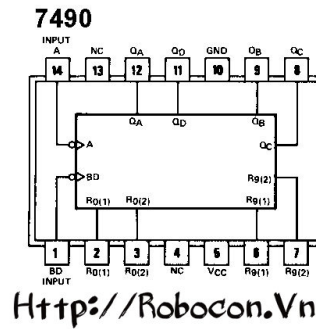
IC 7490 chứa 2 bộ độ MOD gồm MOD 2 và MOD 5. Bộ đếm được bắt đầu đếm từ 0000 đến 1001 và sau đó nó sẽ reset lại giá trị.

Việc reset tự động làm cho bộ đếm sẽ bắt đầu đếm từ 0 và hết thức ở số 9 trong hệ thập phân.



Trong IC 7490 có đến 4 chân reset, nhưng chân này có thể giúp cho IC đếm bằng cách tác động kích hoạt 2 trong 4 chân. IC giao tiếp TTL có thể kết với với các loại bộ đếm và IC có giao tiếp TTL khác.

- Sơ đồ chân của IC 7490:



[Http://Robocon.Vn](http://Robocon.Vn)

Chân	Mô tả chi tiết
Chân 1	Chân 1 là đầu vào xung clock của MOD 5 trong IC. Là chân tích cực mức thấp (chỉ kích hoạt khi có giá trị logic mức thấp), được thay đổi trạng thái ở 3 bit đầu ra. Tại các đầu ra có xung thay đổi từ mức cao đến thấp thì ba bit đầu ra sẽ bị thay đổi.
Chân 2	Chân 2 được sử dụng như một chân reset trong IC. Nó sẽ cho giá trị lớn nhất ở đầu ra. Sử dụng kết hợp với chân 3.
Chân 3	Chân 3 cũng được sử dụng như một chân reset trong IC. Nó sẽ kích giá trị lớn nhất ở đầu ra. Sử dụng kết hợp với chân 2.
Chân 4	Chân 4 sử dụng để hình dáng để nhìn cho mạch PCB. Không quan trọng nó được đầu nối hay không vì nó sẽ không ảnh hưởng đến mạch.
Chân 5	Chân 5 là chân đầu vào cấp nguồn.
Chân 6	Chân 6 được sử dụng như một chân reset trong IC. Nó sẽ xóa tất cả giá trị các đầu ra khi kết hợp với R4.
Chân 7	Chân 7 cũng được sử dụng như một chân reset. Nó sẽ xóa tất cả các đầu ra khi kết hợp với R3.
Chân 8	Chân 8 là một chân đầu ra. Nó là bit thứ hai của dữ liệu đầu ra 4 bit.
Chân 9	Chân 9 cũng là một chân đầu ra. Nó là bit LSB thứ hai (Bit có trọng số thấp thứ 2) của dữ liệu đầu ra 4 bit.
Chân 10	Chân 10 là chân nối đất.
Chân 11	Chân 11 là bit đầu ra có trọng số lớn nhất của dữ liệu đầu ra 4 bit.
Chân 12	Chân 12 là bit đầu ra có trọng số nhỏ nhất của dữ liệu đầu ra 4 bit.
Chân 13	Chân 13 là chân không cần đầu nối. Nó sẽ không ảnh hưởng đến mạch như chân 4.
Chân 14	Chân 14 là chân đầu vào xung clock dùng để cấp xung clock cho MOD 2 của IC.

- Thông số kỹ thuật của IC 7490:

<b>Datasheet</b>	<b>7490</b>
Nguồn đầu vào	4,75 – 5,25V
Nhiệt độ hoạt động	0 – 70 độ C
	Mức CAO nhỏ nhất 2 V
Dây điện áp đầu vào IC	Mức THẤP cao nhất 0,7 V
	Ở mức CAO -0,4mA
Dòng đầu ra của IC	Ở mức THẤP là 8mA
Dây bảo vệ diode bên trong	1,5 V

## Nguyên lý hoạt động của IC 7490

- IC 7490 với cấu trúc gồm 4 Flip Flop, Flip Flop đầu tiên sử dụng MOD 2 và 3 chân còn lại sẽ sử dụng ở MOD 5.
- 2 chân đầu vào sẽ sử dụng xung nhịp để đổi trạng thái đầu ra, các chân reset đều được điều khiển thông qua các cổng AND.

- IC 7490 có đến 4 chân reset với 2 chân xung nhịp và 4 chân đầu ra. Thông thường trước khi sử dụng IC thì ta cần phải hiểu về các chân reset.
- 4 chân reset này đều được sử dụng để có thể điều khiển các chân đầu ra. 4 chân reset này sẽ tạo ra được hơn 16 tổ hợp nhưng trong một số tổ hợp sẽ có các đầu ra nhất định.

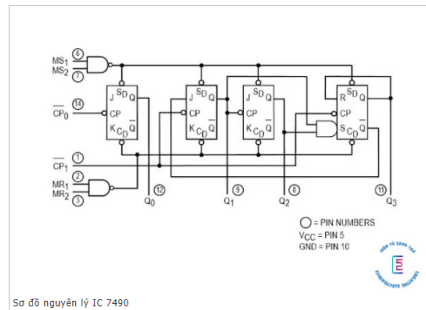


Figure 10: Sơ đồ nguyên lý của IC 7490

### Bảng trạng thái hoạt động của IC 7490

- Chân 1 là chân xung nhịp thứ 2 sẽ được nối với bit có trọng số thấp nhất của IC để có thể hoạt động ở vùng MOD 2 và MOD 5. Chân 14 là chân xung nhịp đầu tiên sẽ được sử dụng để có thể đưa tín hiệu đầu vào xung nhịp cho IC.
- Ở các mức thay đổi trạng thái từ CAO xuống THẤP thì đầu ra sẽ bị thay đổi. Nhưng bạn cần phải luôn ghi nhớ nguyên lý hoạt động của các chân reset của IC, nếu không IC sẽ chạy theo các giá trị ngẫu nhiên hoặc có trường hợp không thay đổi trạng thái đầu ra.
- Bất cứ trường hợp nào chúng ta cho xung nhịp vào IC sẽ cho ra trạng thái đầu ra dạng nhị phân. Mọi số nhị phân sẽ được đại diện cho một số thập phân. Dưới đây là bảng nhị phân tương ứng với các con số thập phân ở đầu ra.

Các chân reset					Đầu ra			
MR1	(R4)MR2	(R3)MS1	(R2)MS2	(R1)Q	DQ	Q2	Q3	BQ A
H	H	L	X	L	L	L	L	L
H	H	X	L	L	L	L	L	L
X	X	H	H	H	L	L	L	H
L	X	L	X	X	ĐẾM			
X	L	X	L	L	ĐẾM			
L	X	X	L	L	ĐẾM			
X	L	L	X	X	ĐẾM			

Figure 11: Bảng trạng thái của IC 7490

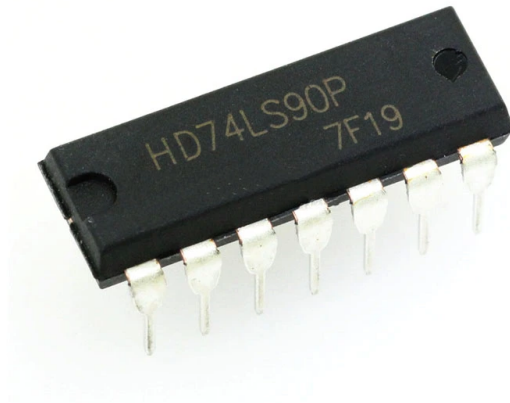


Figure 12: Hình ảnh của IC 7490

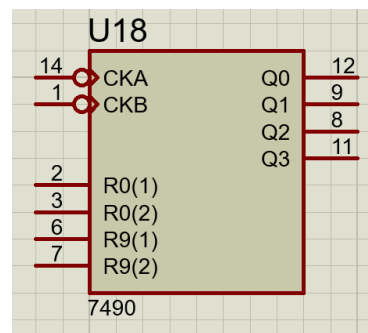


Figure 13: Schematic IC 7490 trong Proteus

#### 4.1.3 IC chốt dữ liệu 74HC573

**IC 74HC573** là 1 vi mạch chốt dữ liệu. Đầu vào của 74HC574 tương thích với đầu ra chuẩn CMOS, trở kéo, bên cạnh đó IC tương thích với đầu ra LS/ALSTTL.

##### Thông số kỹ thuật:

- Điện áp cung cấp: 0.5 – 7V.
- Dòng điện vào mỗi pin: 20mA.
- Dòng điện ra mỗi pin: 35mA.

**Nguyên lý hoạt động của IC 74H573:** Cho dữ liệu vào 8 chân đầu vào từ D0 -> D7 khi chân chốt LE chưa hoạt động hay ở mức thấp thì trạng thái đầu ra chưa thay đổi. Khi chân LE lên mức cao thì dữ liệu đầu ra Q0 -> Q7 sẽ ứng dữ liệu đầu vào tại thời điểm chốt (tạo ra 1 xung). Sau thời điểm chốt dữ liệu đầu ra sẽ không thay đổi khi dữ liệu đầu vào có thay đổi như thế nào đi nữa. Tức là chỉ khi chân chốt LE hoạt động thì dữ liệu đầu ra sẽ thay đổi theo dữ liệu đầu vào tại thời điểm ấy. Còn khi LE không hoạt động thì dữ liệu đầu ra sẽ mãi ở trạng thái đó dù đầu vào thay đổi.

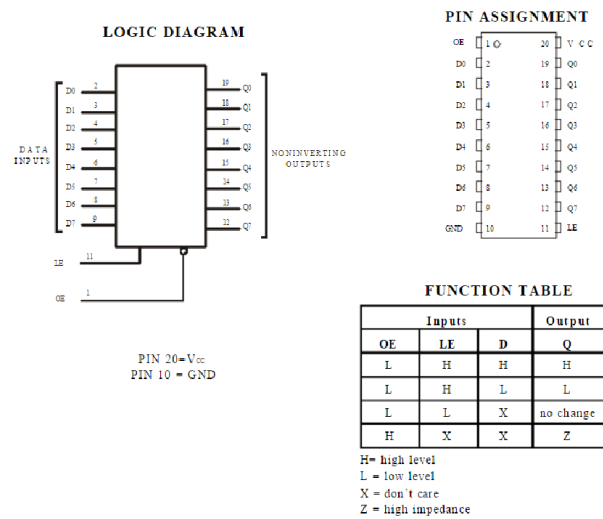


Figure 14: Sơ đồ chân của IC74HC573



Figure 15: Mô tả chi tiết các chân của IC74HC573

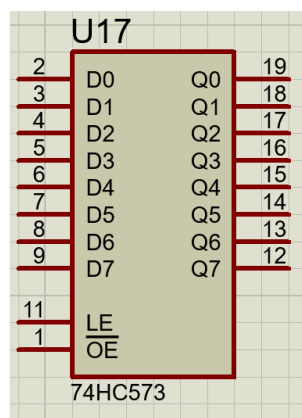


Figure 16: Schematic IC 74HC573 trong Proteus

## 4.2 Khối xử lý

### 4.2.1 IC 74LS83

**IC 74LS83** là một full adder nhị phân 4 bit tốc độ cao. Nó chấp nhận hai từ nhị phân 4 bit ( $A_1 - A_4$ ,  $B_1 - B_4$ ) và một đầu vào mang ( $C_0$ ). Nó tạo ra các đầu ra sum nhị phân ( $\Sigma_1 - \Sigma_4$ ) và đầu ra mang ( $C_4$ ) từ bit quan trọng nhất. Đầu ra của IC luôn ở dạng TTL giúp dễ dàng làm việc với các thiết bị TTL và vi điều khiển khác. IC 74LS83 có kích thước nhỏ và tốc độ nhanh nên đáng tin cậy trong mọi loại thiết bị.

**Sơ đồ chân của IC 74LS83:**

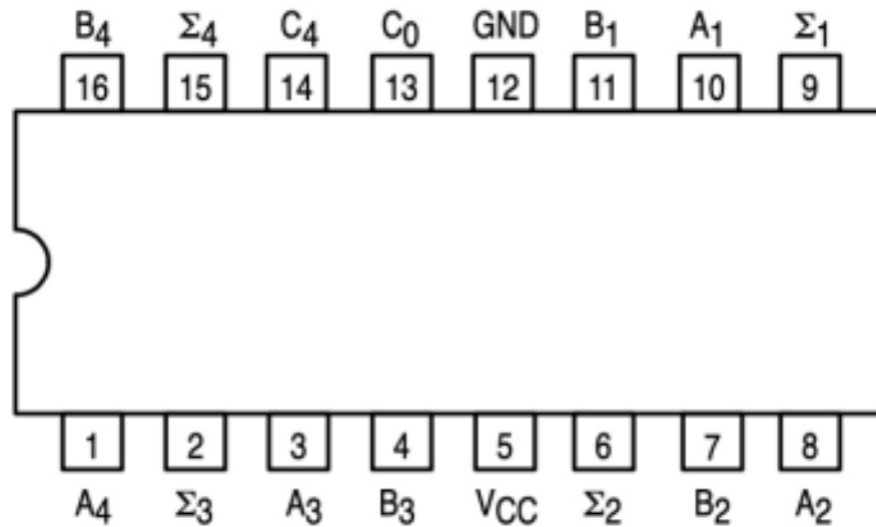


Figure 17: Sơ đồ chân IC 74LS83

### Bảng chức năng cho các chân IC 74LS83:

Số chân	Tên chân	Mô tả
1	A4	Chân đầu vào cho mạch Adder (Bốn chân cho bốn giai đoạn)
2	$\Sigma$ 3	Chân đầu ra cho mạch Adder (Bốn chân cho bốn giai đoạn)
3	A3	Chân đầu vào cho mạch Adder
4	B3	Đầu vào thứ hai cho mạch Adder (Bốn chân cho bốn giai đoạn)
5	VCC	nguồn
6	$\Sigma$ 2	Chân đầu ra cho mạch Adder
7	B2	Đầu vào thứ hai cho mạch Adder
8	A2	Chân đầu vào cho mạch Adder
9	$\Sigma$ 1	Chân đầu ra cho mạch Adder <small>CHÂN LƯU KẾT QUẢ - SẴN SẴN VIẾT VÀO TỶ TRỌNG LẠI</small>
10	A1	Chân đầu vào cho mạch Adder
11	B1	Đầu vào thứ hai cho mạch Adder
12	GND	Chân ground
13	C0	Chân carry - từ giai đoạn trước nếu có.
14	C4	Chân carry out nối tăng với các IC khác.
15	$\Sigma$ 4	Chân đầu ra cho mạch Adder
16	B4	Đầu vào thứ hai cho mạch Adder

Figure 18: Bảng chức năng các chân IC 74LS83

### Hình ảnh của IC 74LS83:

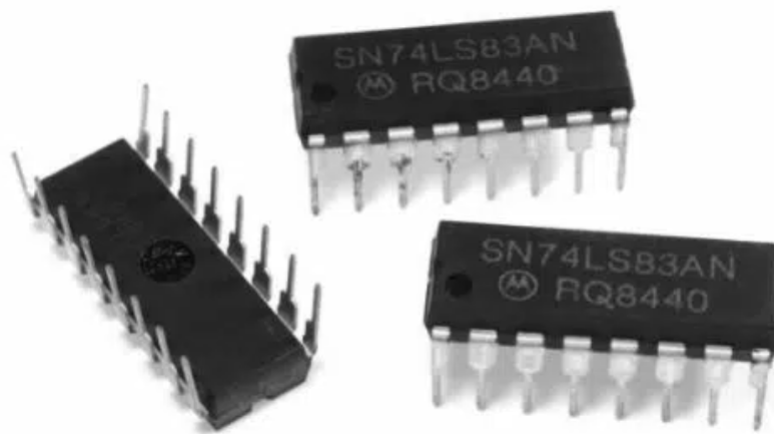


Figure 19: Hình ảnh của IC 74LS83

### Schematic IC 74LS83 trong proteus:

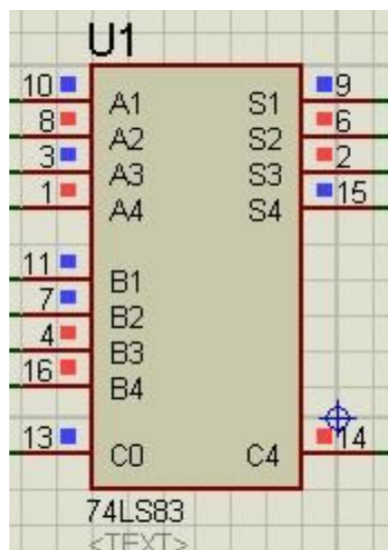


Figure 20: Schematic IC 74LS83 trong proteus

### Thông số kỹ thuật của IC 74LS83:

- Điện áp hoạt động danh nghĩa: 5V.
- Điện áp hoạt động tối đa: 5.5V.
- Độ trễ truyền đầu ra: 16nS.
- Điện áp thấp đầu vào tối đa: 0.8V
- Điện áp cao đầu vào tối thiểu: 2V
- Có các gói PDIP, CDIP, SOIC, TSSOP 16 chân.

### Ứng dụng của IC 74LS83:

- Các phép toán số học tốc độ cao.
- Máy tính thu nhỏ.
- Bộ đếm.
- Thiết kế điều khiển logic đơn giản.

#### 4.2.2 IC 74LS386

**IC 74LS386** chứa bốn cổng OR (XOR) logic dương độc lập 2 đầu vào. Nó là một phần của dòng vi mạch 74XXYY. IC 74LS86 có nhiều điều kiện làm việc và giao tiếp trực tiếp với CMOS, NMOS và TTL. Đầu ra của IC luôn ở dạng TTL giúp dễ dàng làm việc với các thiết bị TTL và vi điều khiển khác. IC 74LS86 có kích thước nhỏ và tốc độ nhanh nên đáng tin cậy trong mọi loại thiết bị.

Sơ đồ chân của IC 74LS386:

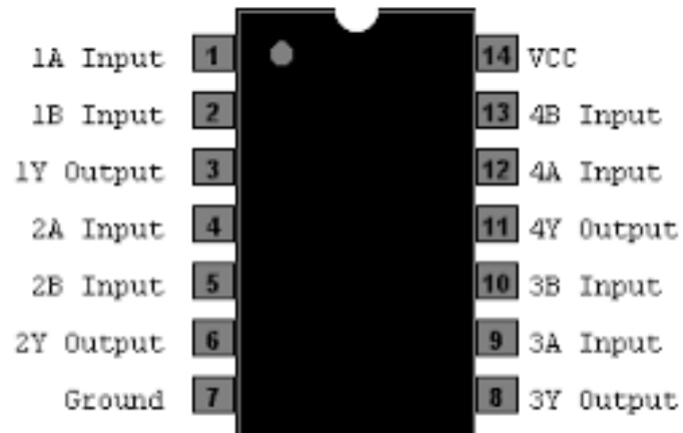


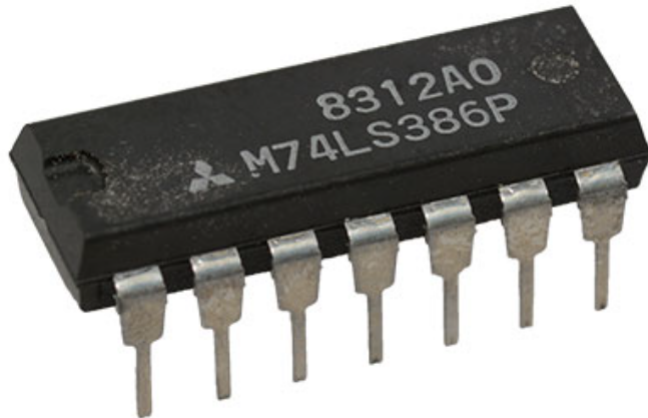
Figure 21: Sơ đồ chân của IC 74LS386

Bảng chức năng cho các chân IC 74LS86:

Số chân	Tên chân	Mô tả
1	1A	Đầu vào 1 của cổng 1
2	1B	Đầu vào 2 của cổng 1
3	1Y	Đầu ra của cổng 1
4	2A	Đầu vào 1 của cổng 2
5	2B	Đầu vào 2 của cổng 2
6	2Y	Đầu ra của cổng 2
7	GND	ground
8	3Y	Đầu ra của cổng 3
9	3B	Đầu vào 2 của cổng 3
10	3A	Đầu vào 1 của cổng 3
11	4Y	Đầu ra của cổng 4
12	4B	Đầu vào 2 của cổng 4
13	4A	Đầu vào 1 của cổng 4
14	VCC	nguồn

Figure 22: Bảng chức năng cho các chân IC 74LS386



**Hình ảnh của IC 74LS386:**

**Figure 23:** Bảng chức năng cho các chân IC 74LS386

**Thông số kỹ thuật của IC 74LS386:**

- Họ công nghệ: LS.
- VCC (Tối thiểu): 4,75V.
- VCC (Tối đa): 5,25V.
- Số kênh: 4.
- Đầu vào mỗi kênh: 2.
- IOL (Tối đa): 16mA.
- IOH (Tối đa) -0,8mA.
- Loại đầu vào: TTL-CMOS tương thích.
- Loại đầu ra: Đẩy-Kéo.
- Các tính năng: Tốc độ cao (TPD 10-50ns).
- Tốc độ dữ liệu (Tối đa): 35Mbps.
- Phạm vi nhiệt độ hoạt động: 0°C đến 70°C.

**Ứng dụng của IC 74LS386:**

- Cổng XOR được sử dụng để thực hiện phép cộng nhị phân trong máy tính.
- Cũng được sử dụng trong thiết bị mạng.
- Ti vi và hệ thống giải trí gia đình.

## 5 Source code dùng trong Đề tài

- Để hiện thực việc nhập 8 bits hay 4 bits cho các tín hiệu Input và xuất Output ở đầu ra và hiện trên LED 7 đoạn, nhóm Chúng Em có thực hiện một đoạn code và sau xuất file .hex để nạp vào vi điều khiển 8051.

- Source code thực hiện:

```
1 #include <stdio.h>
2 #include <REGX51.h>
3 #include <stdlib.h>
4 #include <math.h>
5
6 #define led1 P2_0
7 #define led2 P2_1
8 #define led_on 0
9 #define led_off 1
10
11 char so[14]={0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10,0x89,0x06,0xC7,0x40};
12 int count;
13
14 int BinToDec (binaryNumber)
15 {
16     int p = 0;
17     int decNumber = 0;
18     while (binaryNumber > 0) {
19         decNumber += (binaryNumber % 10) * pow(2, p);
20         ++p;
21         binaryNumber /= 10;
22     }
23     return decNumber;
24 }
25
26 void delay (int time) {
27     while(time--);
28 }
29
30 void main() {
31     int num1,
32         num2,
33         num3,
34         num4,
35         num5,
36         num6,
37         num7,
38         num8,
39         sogoc,
40         chuc,
41         donVi;
42     led1 = led2 = led_off;
43     while(1) {
44         num1 = P1_7;
45         num2 = P1_6;
46         num3 = P1_5;
47         num4 = P1_4;
48         num5 = P1_3;
49         num6 = P1_2;
50         num7 = P1_1;
51         num8 = P1_0;
52
53         sogoc = 0;
54         sogoc=num1*10000000+num2*1000000+num3*100000+num4*10000+num5*1000+num6*100+num7
55             *10+num8;
56         count = BinToDec(sogoc);
57         chuc = count/10;
58         donVi = count%10;
59         led1 = led_on;
60         P0 = so[chuc];
61         delay(3000);
62         led1 = led_off;
```

```

63  led2 = led_on;
64  P0 = so[donVi];
65  delay(3000);
66  led2 = led_off;
67  }
68  }

```

Code 1: Source Code

## 6 Hiện thực các mạch thành phần

### 6.1 Hiện thực mạch chia

Để có thể thực hiện mạch chia một cách có hệ thống và xúc tích, nhóm em xây dựng 3 khối mạch con chính:

#### 6.1.1 Khối S

**Input** : 4 bit của số bị trừ (A1-A4) và 4 bit của số trừ (B1-B4)

**Output** : 4 bit hiệu (S1-S4) và bit nhớ *Cout*

**Chức năng** : Thực hiện phép trừ 2 số 4 bit theo nguyên tắc cộng với bù 2 của số trừ

Vì phép toán đầu tiên luôn là phép trừ nên một chân input vào cổng XOR luôn là 1 để thực hiện việc lấy bù 2 của số trừ (các cổng XOR đảo các bit của số trừ) và chân  $C_0$  sẽ là 1.

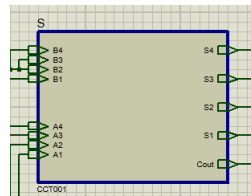


Figure 24: Biểu diễn Khối S

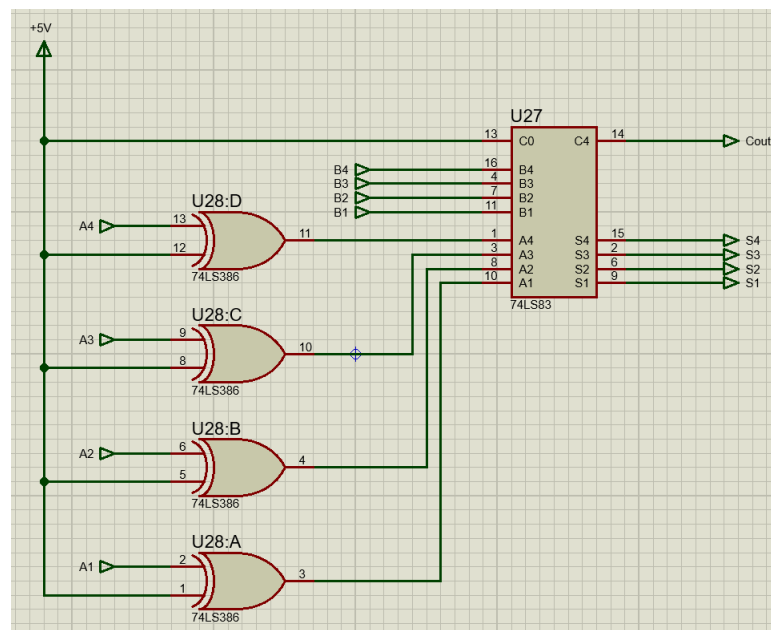


Figure 25: Sơ đồ nguyên lý Khối S

### 6.1.2 Khối $SUB_i$

**Input** : 4 bit của số bị trừ (A1-A4), 4 bit của số trừ (B1-B4) và bit nhớ  $C_{in}$

**Output** : 4 bit hiệu (S1-S4) và bit nhớ  $C_{out}$

**Chức năng** : Thực hiện phép trừ 2 số 4 bit theo nguyên tắc cộng với bù 2 của số trừ có kèm theo bit nhớ từ kết quả giai đoạn trước

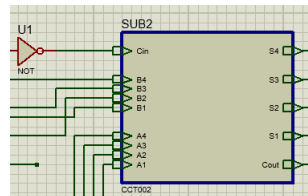


Figure 26: Biểu diễn Khối  $SUB_i$

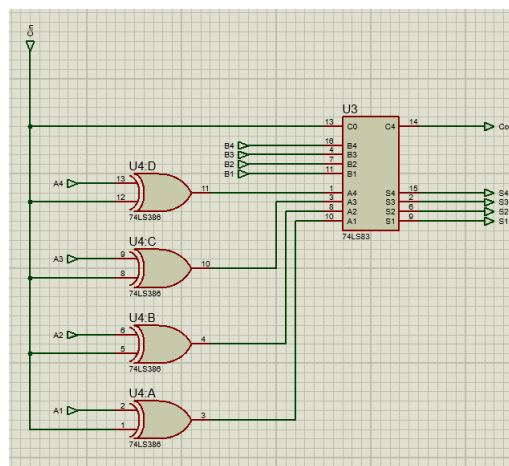


Figure 27: Sơ đồ nguyên lý Khối  $SUB_i$

### 6.1.3 Khối Remainder Modify

**Input** : 4 bit của số bị trừ (A1-A4), 4 bit của số trừ (B1-B4)

**Output** : 4 bit hiệu (S1-S4)

**Chức năng** : Thực hiện phép cộng số dư với số chia nếu số dư là âm, số dư không âm thì mạch giữ nguyên số dư

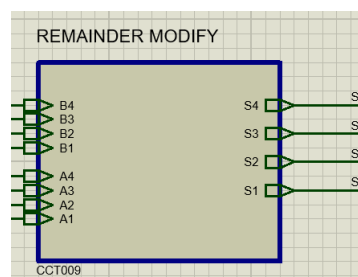


Figure 28: Biểu diễn Khối Remainder Modify

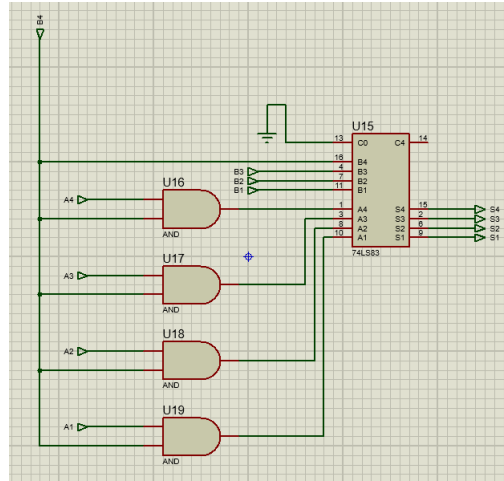


Figure 29: Sơ đồ nguyên lý Khối Remainder Modify

#### 6.1.4 Kết hợp thành mạch chia hoàn chỉnh

Dựa trên thiết kế của giải thuật chia, ta có thể sắp xếp các khối đã nêu trên thành một mạch chia hoàn chỉnh. Trong project, nhóm có làm thêm một chức năng báo nếu số chia là 0. Tuy kết quả vẫn sẽ được hiển thị nhưng output Divide\_by\_0 sẽ sáng lên báo hiệu phép tính có lỗi và kết quả không đúng.

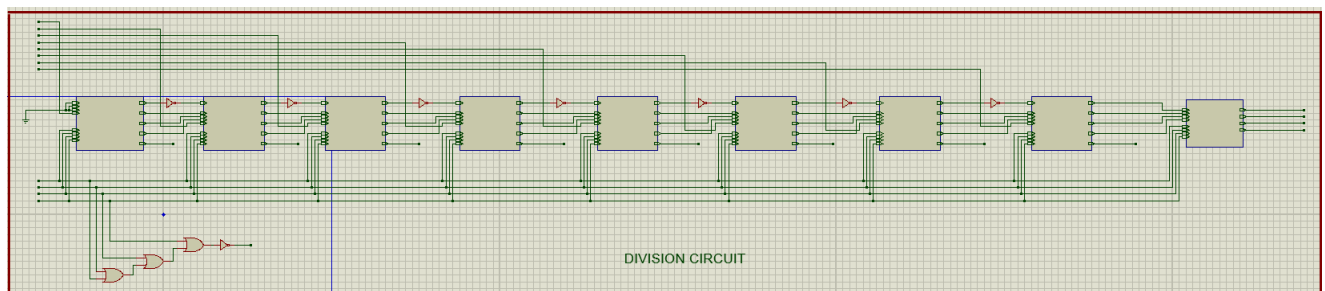


Figure 30: Mạch chia hoàn chỉnh 8bit và 4bit

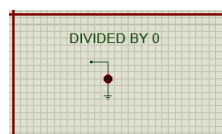


Figure 31: Chức năng phát hiện chia cho 0

## 6.2 Hiện thực mạch trừ

Để hiện thực được mạch trừ, số bị trừ và số trừ phải có số bit bằng nhau. Vì số bị trừ có số bit lớn hơn (8 bit) nên số trừ sẽ được mở rộng ra 8 bit bằng cách thêm các bit 1 vì ta đang biểu diễn số chia theo dạng bù 2. Nhóm sử dụng 2 adder 4-bit theo cấu trúc xếp tầng. Hình dưới là mạch trừ hoàn chỉnh cho số bị trừ 8-bit và số trừ 4-bit :

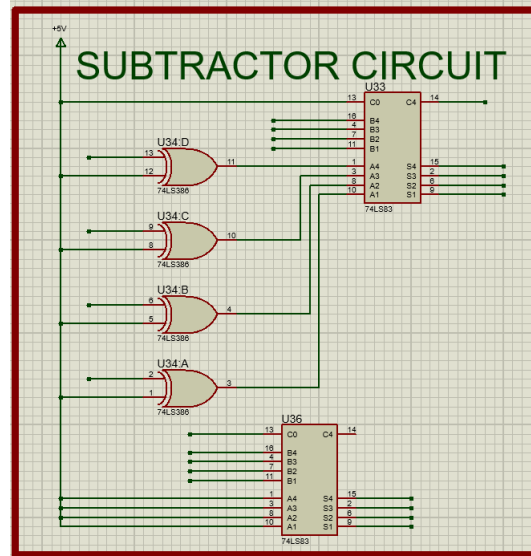
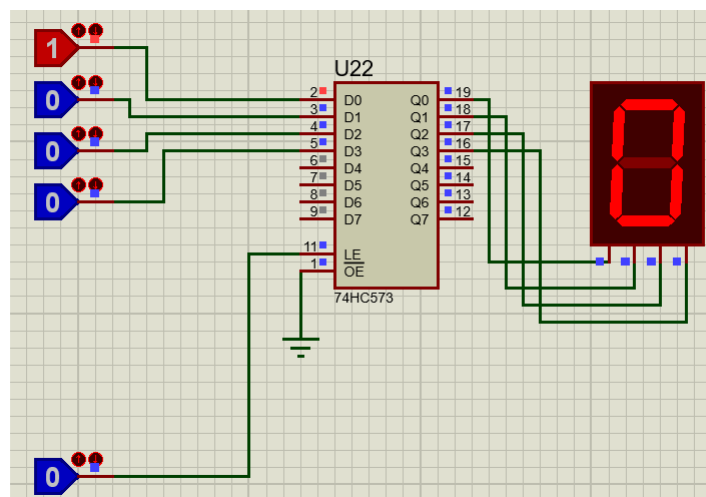


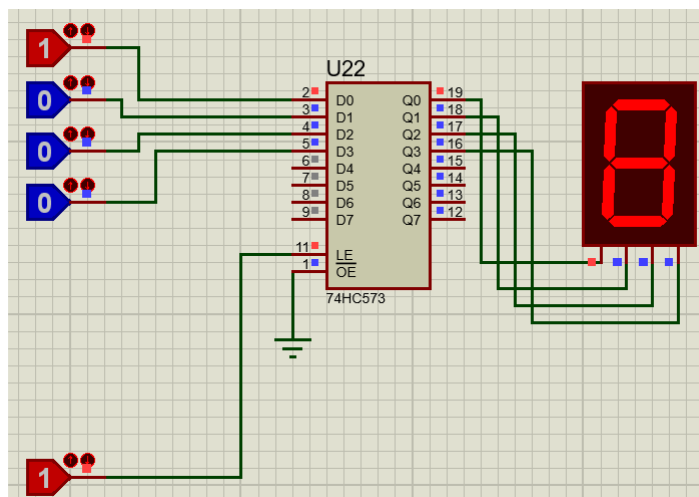
Figure 32: Mạch trừ hoàn chỉnh

### 6.3 Hiện thực mạch lưu

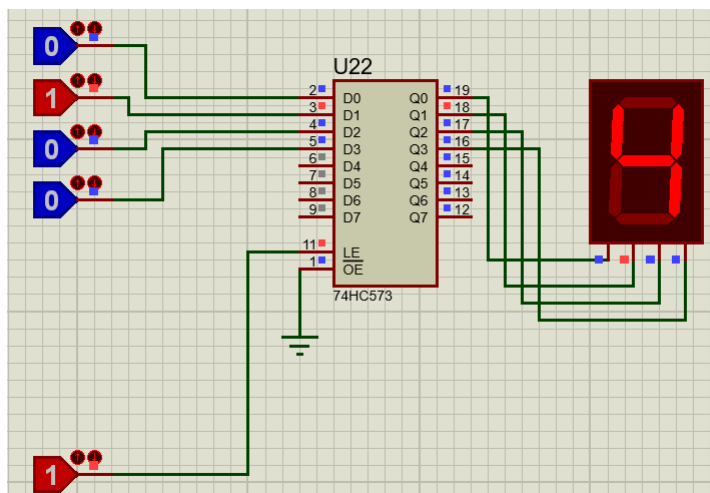
- Để hiện thực mạch lưu dữ liệu trong đề tài này, nhóm Chúng Em sử dụng IC chốt dữ liệu 74HC573 với mục đích lưu dữ liệu sau mỗi bước xử lý. Các IC chốt sẽ được phân bổ đều đối với từng chức năng nhập và xuất kết quả trong toàn mạch.
- Nhắc lại về nguyên lý hoạt động của IC 74HC573: Cho dữ liệu vào 8 chân đầu vào từ D0 -> D7 khi chân chốt LE chưa hoạt động (tín hiệu LE = 0) hay ở mức thấp thì trạng thái đầu ra chưa thay đổi. Khi chân LE lên mức cao (tín hiệu LE = 1) thì dữ liệu đầu ra Q0 -> Q7 sẽ ứng dữ liệu đầu vào tại thời điểm chốt (tạo ra 1 xung). Sau thời điểm chốt dữ liệu đầu ra sẽ không thay đổi khi dữ liệu đầu vào có thay đổi như thế nào đi nữa. Tức là chỉ khi chân chốt LE hoạt động thì dữ liệu đầu ra sẽ thay đổi theo dữ liệu đầu vào tại thời điểm ấy. Còn khi LE không hoạt động thì dữ liệu đầu ra sẽ mãi ở trạng thái đó dù đầu vào thay đổi.
- Ví dụ: khi ở trạng thái ban đầu, với 4 bits Input là 1000 và tín hiệu chân LE = 0 thì tín hiệu Output không xuất dữ liệu 1000 (= 8) do IC đang hoạt động chốt dữ liệu.



Khi kích hoạt tín hiệu chân LE = 1, Output sẽ là 1000, và số 8 sẽ được hiển thị trên LED 7 đoạn.



Lúc này, mọi thay đổi ở Input sẽ kéo theo mọi tín hiệu đầu ra ở Output. Ta có thể thử với 4 bits 0100, LED 7 đoạn sẽ hiện số 4.



## 7 Tổng hợp các mạch thành phần

Từ việc hiện thực những mạch đơn nêu trên, nhóm tổng hợp các mạch con lại thành một mạch hoàn chỉnh bao gồm chia, trừ và lưu kết quả

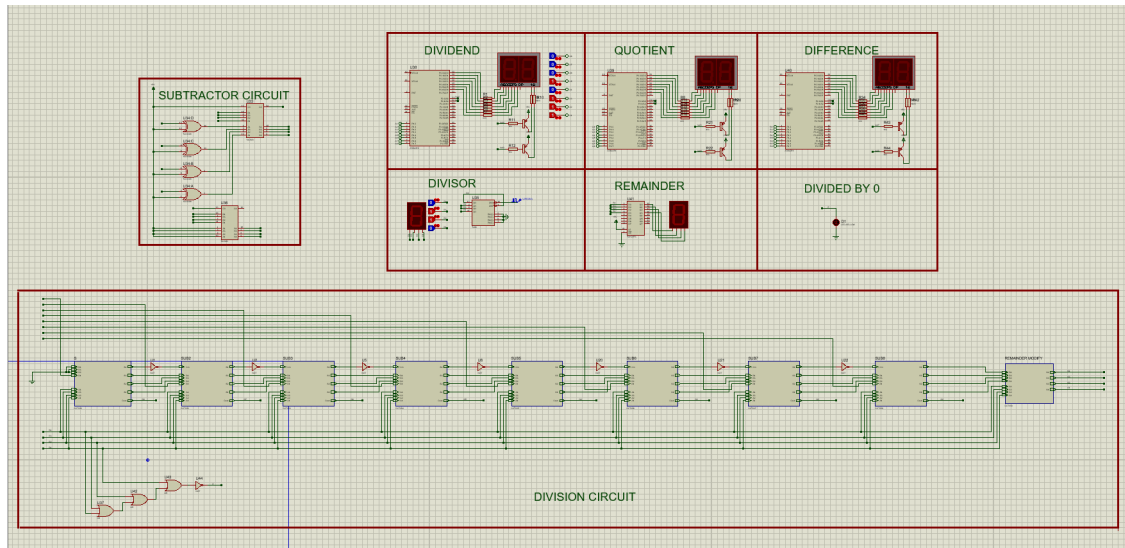


Figure 33: Mạch hoàn chỉnh với các chức năng cơ bản

## 8 Kiểm thử chức năng

### 8.1 Ví dụ 1

Số bị trừ là  $00001111_2$  (15), số trừ là  $0110_2$  (6)

- Thực hiện phép chia :
  - Thương =  $10_2$  (2)
  - Số dư =  $11_2$  (3)
- Thực hiện phép trừ : hiệu là  $00001001_2$  (9)

Kết quả :

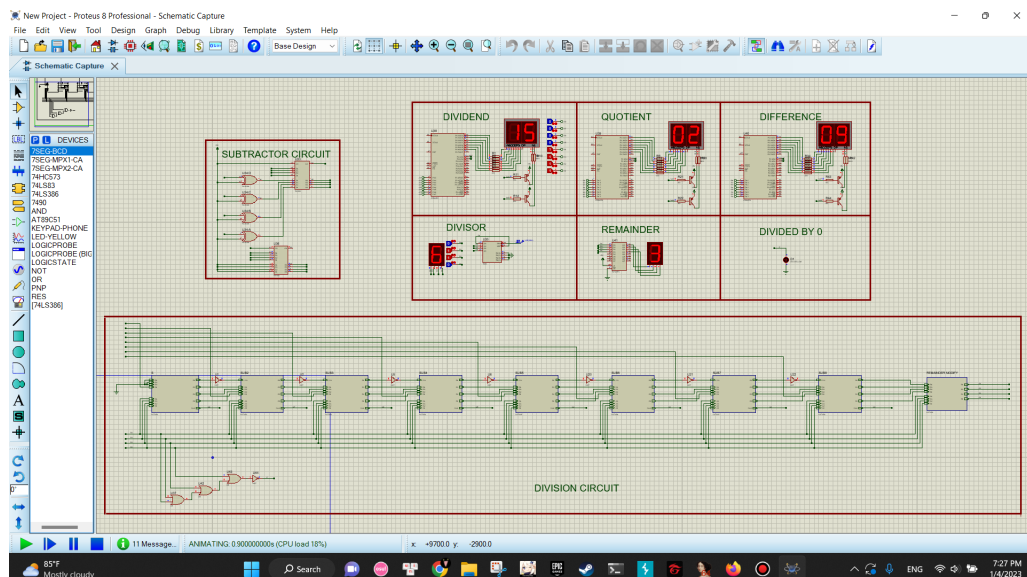


Figure 34: Kết quả mô phỏng ví dụ 1 trên Proteus

### 8.2 Ví dụ 2

Số bị trừ là  $00011101_2$  (29), số trừ là  $0100_2$  (4)



- Thực hiện phép chia :
  - Thương =  $00000111_2$  (7)
  - Số dư =  $0001_2$  (1)
- Thực hiện phép trừ : hiệu là  $00011001_2$  (25)

Kết quả :

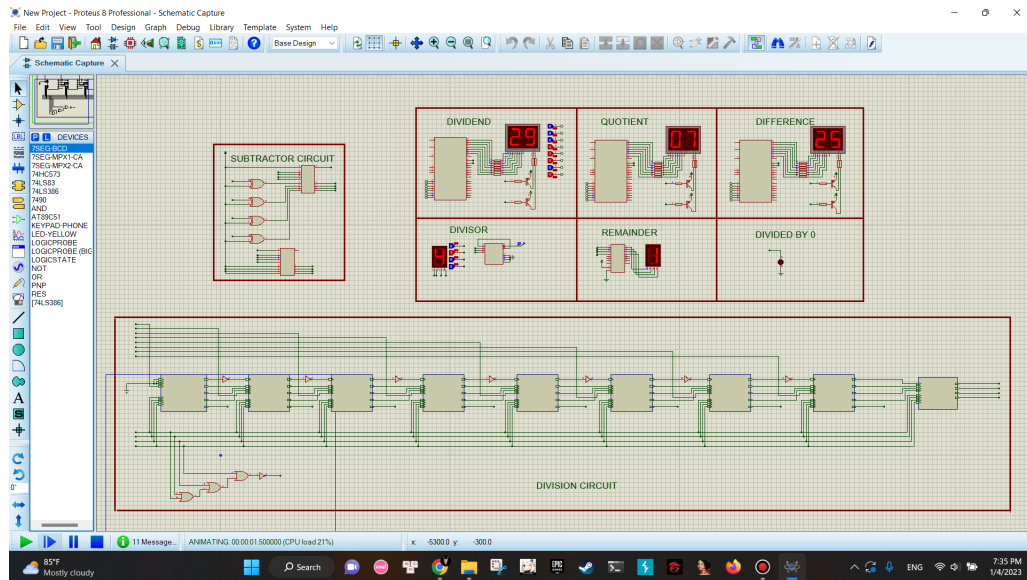


Figure 35: Kết quả mô phỏng ví dụ 2 trên Proteus

### 8.3 Ví dụ 3

Số bị trừ là  $00000000_2$  (0), số trừ là  $0000_2$  (0)

- Thực hiện phép chia :
  - Thương = *error*
  - Số dư = *error*
  - Vì phép chia có số chia là 0 nên xảy ra lỗi *divide\_by\_0*
- Thực hiện phép trừ : hiệu là  $00000000_2$  (0)

Kết quả :

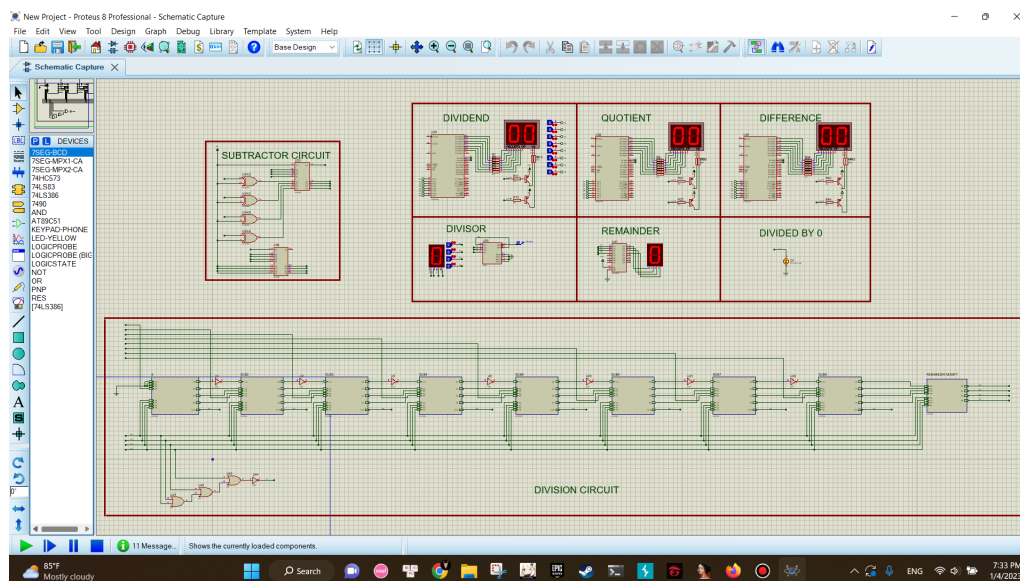


Figure 36: Kết quả mô phỏng ví dụ 3 trên Proteus

## 9 Đánh giá

### 9.1 Ưu điểm

- Sản phẩm đã hoạt động bình thường và đáp ứng đủ các yêu cầu cơ bản và nâng cao : mạch trừ và mạch chia và có nhớ kết quả.
- Nhờ đề tài mà các thành viên trong nhóm đã có thể tiếp cận được các giải thuật, các mạch, chip và kiến thức cơ bản về ứng dụng vi xử lý - vi điều khiển.
- Hệ thống mạch nhóm Chúng Em xây dựng tương đối khá ít linh kiện, tận dụng được chức năng của bộ vi điều khiển 8051 để xử lý cho các tín hiệu Input và Output.

### 9.2 Nhược điểm

- Trong quá trình hiện thực đồ án, nhờ tư vấn của thầy Vũng về việc thiết kế việc nhập các Input từ bàn phím nhưng nhóm vẫn chưa hiện thực thành công theo ý tưởng Thầy tư vấn.
- Quá trình bắt đầu thực hiện đồ án nhóm vẫn còn gặp khó khăn trong việc lên ý tưởng và tìm các tài liệu và các linh kiện phù hợp cho đề tài được giao.
- Vì số chia được biểu diễn có dấu nên chỉ có thể có những giá trị phù hợp là từ 1 đến 8, số chia = 9 thì kết quả sẽ bị sai.
- Mặc dù mô phỏng trên Proteus chạy đúng chức năng và đúng kết quả, vẫn còn gặp những lỗi Internal Exception bất ngờ.

### 9.3 Giải pháp

- Quản lý thời gian tốt hơn và có một kế hoạch rõ ràng hơn
- Để giải quyết việc số chia bằng 9, ta có thể đưa số chia thành 5 bit, do đó số chia sẽ có thể nằm trong khoảng 0-9 (cụ thể hơn là 0-16). Tuy nhiên, nhóm đã không có đủ thời gian để hiện thực giải pháp này

## 10 Phân công công việc và kế hoạch thực hiện đồ án

Kế hoạch thực hiện đồ án :

STT	Tên công việc / sự kiện	Ngày bắt đầu	Ngày kết thúc	Ghi chú
1	Nhận đề tài	9h00, 13/09/2022		
2	Tìm hiểu về lý thuyết và tạo project github	15/09/2022	29/09/2022	Đúng tiến độ
3	Xây dựng mạch chia và mạch trừ	01/10/2022	08/10/2022	Trễ 1 tuần
4	Tìm hiểu về mạch lưu, xây dựng mạch trừ	16/10/2022	01/11/2022	Đúng tiến độ
5	Viết báo cáo cơ sở lý thuyết	02/11/2022	09/11/2022	Đúng tiến độ
6	Báo cáo tiến độ và hỏi ý kiến thầy	11/11/2022		
7	Hiện thực mạch trừ	12/11/2022	19/11/2022	Đúng tiến độ
8	Hiện thực mạch lưu	22/11/2022	01/12/2022	Đúng tiến độ
9	Debug mạch chia và mạch trừ	02/12/2022	29/12/2022	Sớm 1 tuần
10	Debug mạch lưu và tổng hợp mạch	30/12/2022	03/01/2023	Đúng tiến độ
11	Hoàn tất báo cáo	04/01/2023		Đúng tiến độ

Phân công công việc :

Tên	Công việc
Trịnh Cao Thắng	Mạch chia + Mạch trừ + Báo cáo
Nghiêm Lương Sơn	Mạch lưu + Báo cáo

## 11 Tài liệu tham khảo

- Giáo trình Kỹ thuật số - Nguyễn Trung Lập.
- Linh kiện IC 74HC573, truy cập từ <https://lkcg.vn/linh-kien-ic-74hc573-chan-cam-dip20>
- Tìm hiểu về IC 74LS86, truy cập từ <https://dientutuonglai.com/tim-hieu-74ls86.html>