**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**A green pencil with a yellow pencil in the middle

AI-generated content may be incorrect.**

**BÁO CÁO THỰC HÀNH LAB4 &5**

**THIẾT KẾ DATAPATH, CONTROL UNIT**

**MÔN CE118 – THIẾT KẾ LUẬN LÝ SỐ**

**Giảng viên hd: Ths. HỒ NGỌC DIỄM**

Sinh viên thực hiện :

Nguyễn Đình Anh. MSSV: 23520057

Nguyễn Hoàng Quốc Cường. MSSV: 23520200

Lớp CE118.P21

Nội dung thực hành:

Thiết kế một Simple DATAPATH 8-bit để thực hiện phép toán S = 1 + 2 + 3 + … + n Với n là input được nhập từ ngõ vào.

1. Thiết kế mã giả

S = 0

i = 0

n = input.

while ( n> 0 )

i = i + 1;

S = s + i

n = n -1 ;

1. Xác định các thành phần cần thiết của datapath

* Chúng ta cần một tập thanh ghi REGISTER FILE ít nhất 4 địa chỉ để lưu các giá trị S , I , N, với thanh ghi đầu tiên luôn có giá trị 0.
* Một SELECTOR ( Mux2\_1) để chọn lựa ngõ vào là input hoặc là dữ liệu từ ALU.
* Một ALU gồm tối thiểu các chức năng cộng , cộng 1 , trừ 1.
* Một TRI-STATE để xác định thời điểm tính xong kết quả và đưa kết quả ấy ra output.

1. Xây dựng ASM chart.

S0: DONE =0

START

0

1

S1: DATA = INPUT ( R1)

S2: S= 0 (R2)

S4: I = I +1

S3: I =0 (R3)

S5: S = S + I

S6: DATA = DATA -1

1

DATA

!= 0

0

S7: DONE =1

OUTPUT = S

1. Từ lưu đồ trên, ta xây dựng bảng chuyển đổi trạng thái dưới: (Hình 1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STATE | Q2Q1Q0 | START, DATA !=0 | | | |
| 00 | 01 | 10 | 11 |
| S0 | 000 | 000 | 000 | 001 | 001 |
| S1 | 001 | 010 | 010 | 010 | 010 |
| S2 | 010 | 011 | 011 | 011 | 011 |
| S3 | 011 | 100 | 100 | 100 | 100 |
| S4 | 100 | 101 | 101 | 101 | 101 |
| S5 | 101 | 110 | 110 | 110 | 110 |
| S6 | 110 | 111 | 100 | 111 | 100 |
| S7 | 111 | 000 | 000 | 000 | 000 |

Hình 1.

* Từ bảng trên, ta có karmap như sau (Hình 2) .

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q1Q0  START, DATA !=0 | Q2 = 0 | | | | Q2= 1 | | | |
| 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
|  |  |  |  |  |  |  |  |
| 00 | 000 | 010 | 100 | 011 | 101 | 110 | 000 | 111 |
| 01 | 000 | 010 | 100 | 011 | 101 | 110 | 000 | 100 |
| 11 | 001 | 010 | 100 | 011 | 101 | 110 | 000 | 100 |
| 10 | 001 | 010 | 100 | 011 | 101 | 110 | 000 | 111 |

Hình2.

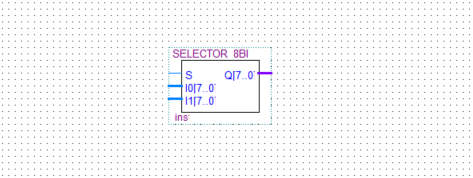
* Từ karmap, ta rút ra được có ngõ vào kế tiếp:

Q2(NEXT) = Q2’Q1Q0 + Q2Q0’ + Q2Q1’

Q1(NEXT) = Q1’Q0 + Q2’Q1Q0’ + (DATA !=0)’Q2Q1Q0’

Q0(NEXT) = Q2’Q1’Q0’START + Q2’Q1Q0’ + Q2Q1’Q0’ + Q2Q1Q0’(DATA!=0)’.

* Bảng điều khiển của các ngõ cần thiết kế:
* SELECTOR : S =1-> Q = I1, S = 0 -> Q = I0 ( Hình 3).



Hình 3. SELECTOR.

* REGISTER FILE: (Hình 4) .

CLK: xung clock

WRI\_ENA: cho phép ghi.

WR\_A1: Địa chỉ ghi dữ liệu.

WR\_A0:

READ\_ENA\_A: Cho phép đọc dữ liệu thứ 1.

R\_A0: Địa chỉ đọc dữ liệu thứ 1

R\_A1

READ\_ENA\_B: cho phép đọc dữ liệu ngõ thứ 2

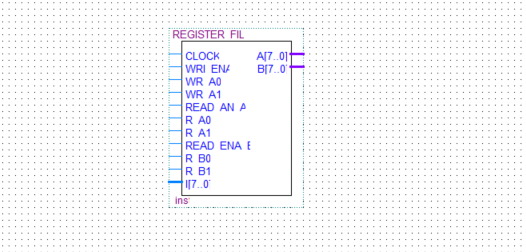
R\_B0: địa chỉ đọc dữ liệu ngõ thứ 2

R\_B1:

I : Dữ liệu được đưa vào.

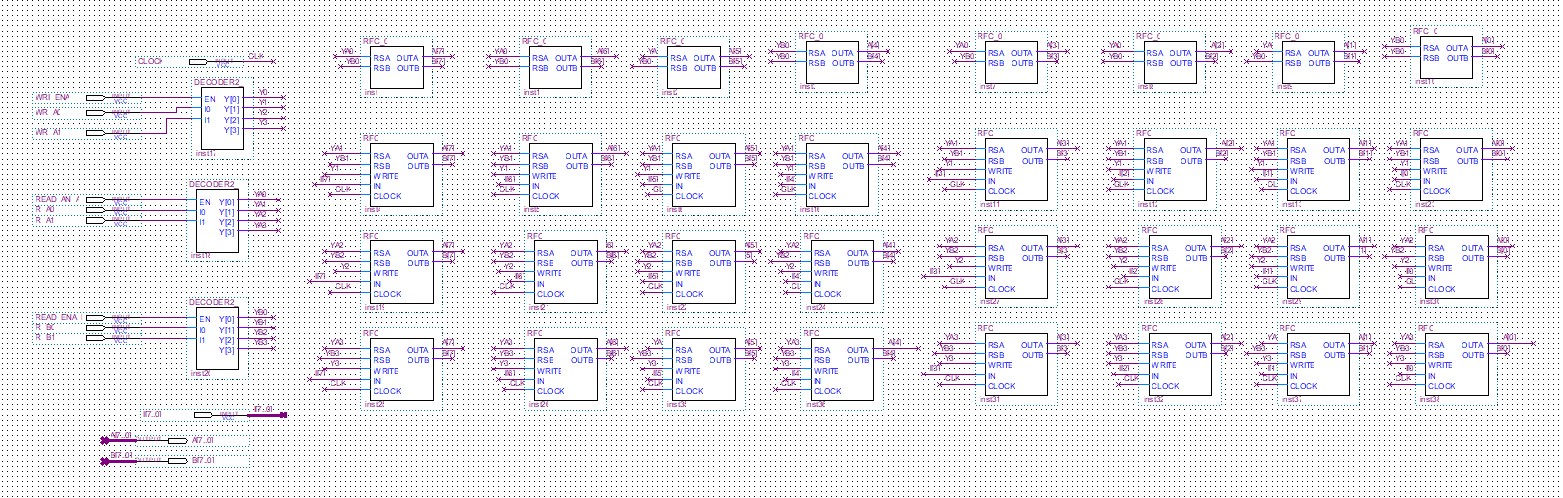
A: Ngõ ra đọc dữ liệu thứ 1

B: Ngõ ra đọc dữ liệu thứ 2.



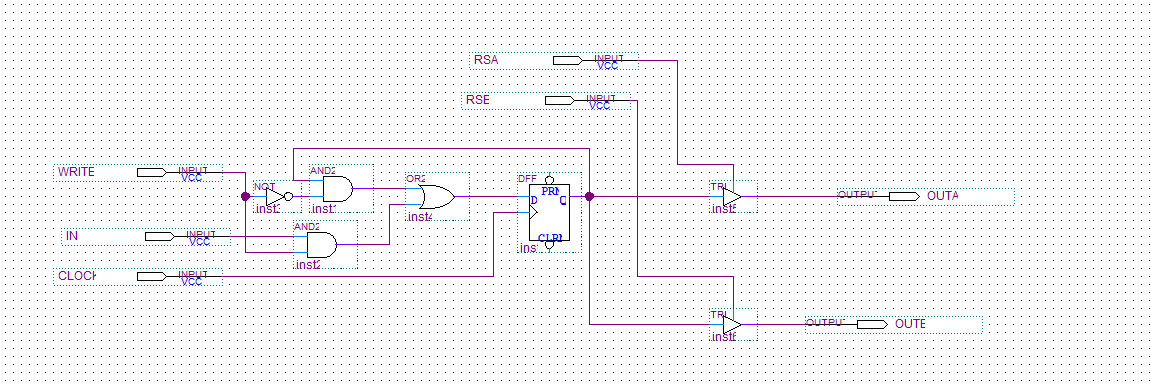
Hình 4. REGISTER FILE.

* Chi tiết thiết kế của REGISTER FILE: (Hình 5).

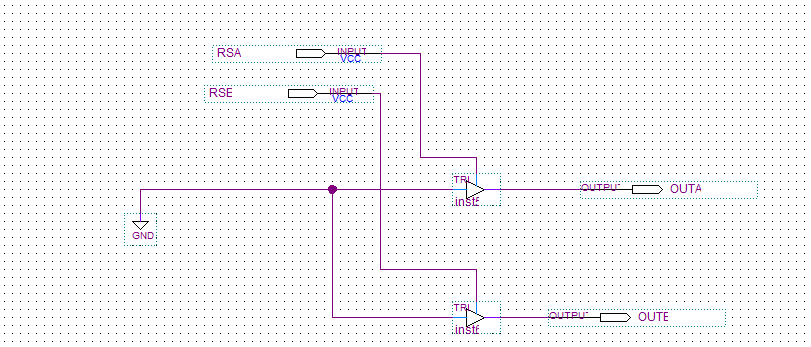


Hình 5. REGISTER FILE 8 bít, 4 địa chỉ.

* Register file được thiết kế bằng cách ghép các khối RFC với nhau, trong đó mỗi khối RFC được thiết kế như sau (Hình 6)

Hình 6.

* Vì thanh ghi đầu tiên luôn giữ giá trị 0, nên ta thiết kế cell này như sau: (Hình 7)



Hình 7.

* Bộ tính toán ALU: ( Hình 8).

S1S0 : 00 -> A + B, 10 -> A +1, 10 -> A + 0, 11 -> A-1 .

A close-up of a diagram

AI-generated content may be incorrect.

Hình 8. Bộ tính toán ALU.

* Buffer : S = 1 -> Q = I, S = 0 -> Q = Z (Hình 9).

A diagram of a graph

AI-generated content may be incorrect.

Hình 9. Buffer.

* Từ các tín hiệu trên, ta có bảng control word như sau ( Hình 10)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CONTROL WORD | | | | | | | | | | | | | |
| State | Wr  Ena | Wr  \_A1 | Wr  \_A0 | R\_  Ena  \_A | R  \_A1 | R  \_A0 | R\_  Ena  \_B | R  \_B1 | R\_B  0 | S1 | S0 | S  (bufer) | S (select) |
|  |  |  |  |  |  |  |  |  |  |  |
| S0 | 0 | X | X | 0 | X | X | 0 | x | X | x | X | 0 | 0 |
| S1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| S2 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| S2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| S4 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | X | X | 0 | 1 | 0 | 0 |
| S5 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| S6 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | X | X | 1 | 1 | 0 | 0 |
| S7 | 0 | X | X | 1 | 1 | 0 | 0 | X | x | 1 | 0 | 1 | 0 |

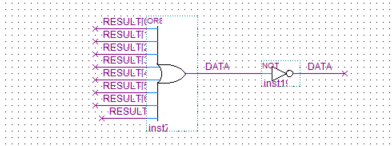
Hình 10.

* Từ bảng control word, ta đưa lên bìa kar và rút gọn được các tín hiệu như sau:
* WR\_ENA = Q2Q1’ + Q2’Q0 + Q1Q0’
* WR\_A1 = Q2Q1’ + Q2’Q1
* WR\_A0 = Q2’Q0 + Q2Q0’
* R\_ENA\_A = Q2+ Q1
* R\_A1 = Q2Q1’ + Q2Q0
* R\_A0 = Q2Q1’ + Q2Q0’
* R\_ENA\_B= Q2’Q1 + Q2Q1’Q0
* R\_B1 = Q2
* R\_B1 = 0
* S1 = Q2Q1
* S0 = Q2Q0’
* S(BUFFER) = DONE = Q2Q1Q0
* S (SELECTOR ) = S1 = Q2’Q1’Q0.
* ( data != 0 ) = or tất cả các bít dữ liệu sau mỗi trạng thái.
* Ta có datapath như sau ( Hình 11)

A diagram of a computer

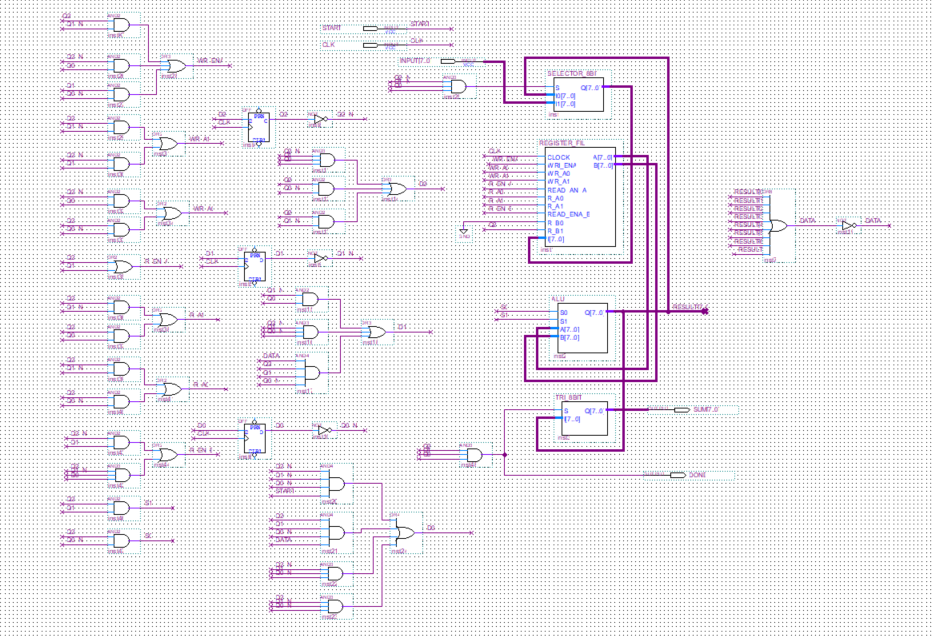
AI-generated content may be incorrect.

Hình 11.



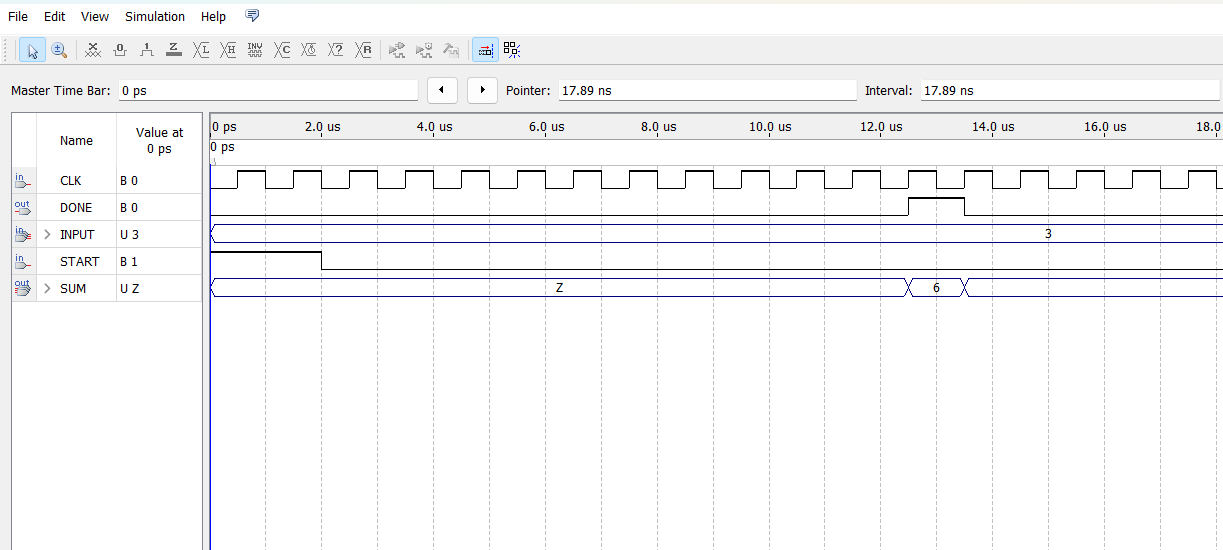
Hình 12. Ngõ (Data ! = 0 ) mà đúng hoặc sai.

* Thiết kế mạch hoàn chỉnh: (Hình 13).



Hình 13.

* Kết quả kiểm tra bằng waveform:



Hình 14.

* Với test case thứ 1: Input là 3, tổng là 0 + 1+ 2 + 3 = 6 -> test case thứ nhất đúng, mất 13 chu kì để hoàn thành. (Hình 14)



Hình 15.

* Với test case thứ 2: Input đưa vào 5, vậy tổng là 1 + 2 + 3 + 4 + 5 = 15 - > test case thứ 2 đã đúng, mất 19 chu kì để hoàn thành. (Hình 15)



Hình 16.

* Với test case thứ 3: Input đưa vào là 5, vậy tổng là 1 + 2 + 3 + 4 + 5 + 6 = 21 -> test case thứ 3 đã đúng, mất 22 chu kì để hoàn thành. (Hình 16).