



LAB 4 & 5: THIẾT KẾ DATAPATH, CONTROL UNIT

Trần Minh Tiến 23521586

Trần Hoàng Thông 23521529

1.1 Mục tiêu

Một mạch chuyên dụng hoặc vi xử lý có thể được thiết kế dựa trên hai khối chính là Controller và Datapath.

Controller có nhiệm vụ điều khiển đường đi của dữ liệu (điều khiển việc đọc ghi của Register File, ...) và điều khiển việc thực hiện tính toán (điều khiển Opcode của khối ALU, ...) trong khối Datapath.

Datapath chứa các khối cần thiết để thực hiện việc tính toán (Register File, ALU, Bộ dịch,...) được điều khiển bởi Controller.

Trong bài Lab này sinh viên sẽ

- Thiết kế một Simple DATAPATH 8-bit để thực hiện phép toán

$$S = 1 + 2 + 3 + \dots + n \text{ Với}$$

n là input được nhập từ ngõ vào.

1.2 Nội dung thực hành

Sinh viên sẽ dựa vào yêu cầu của bài thực hành và các cơ sở lý thuyết để xác định các thành phần cần thiết của DATAPATH. Tiến hành thiết kế Datapath và khối điều khiển Control Unit cho mạch.

1.3 Sinh viên chuẩn bị

1. Viết giải thuật với mã giả (pseudo code) để thực hiện yêu cầu bài lab.
2. Xác định các thành phần cần thiết của DATAPATH
3. Thiết kế schematic cho các thành phần của Datapath (ALU, Register File, ...)

4. Vẽ mạch DATAPATH hoàn chỉnh
5. Xác định lưu đồ chuyển trạng thái của Control Unit
6. Xây dựng máy trạng thái cho khối điều khiển
7. Lập bảng các giá trị ngõ ra (output) của Control Unit để thực hiện giải thuật.

Thực hành: Thiết kế Luận lý Số (CE118)



1.4 Hướng dẫn thực hành

1. Tạo một project mới, đặt tên: E/CE118_lab/lab45_MSSV
2. Kết nối các thành phần đơn lẻ để tạo thành DATAPATH hoàn chỉnh trên phần mềm Quartus II
3. Thiết kế Control Unit hoàn chỉnh trên phần mềm Quartus II
4. Kết nối Datapath và Control Unit để thực hiện giải thuật trên
5. Hiện thực và thử nghiệm trên Kit DE2

Bài làm

Mã giả:

RegSum=0

RegN= N

While(RegN!=0)

 RegSum=RegSum+N

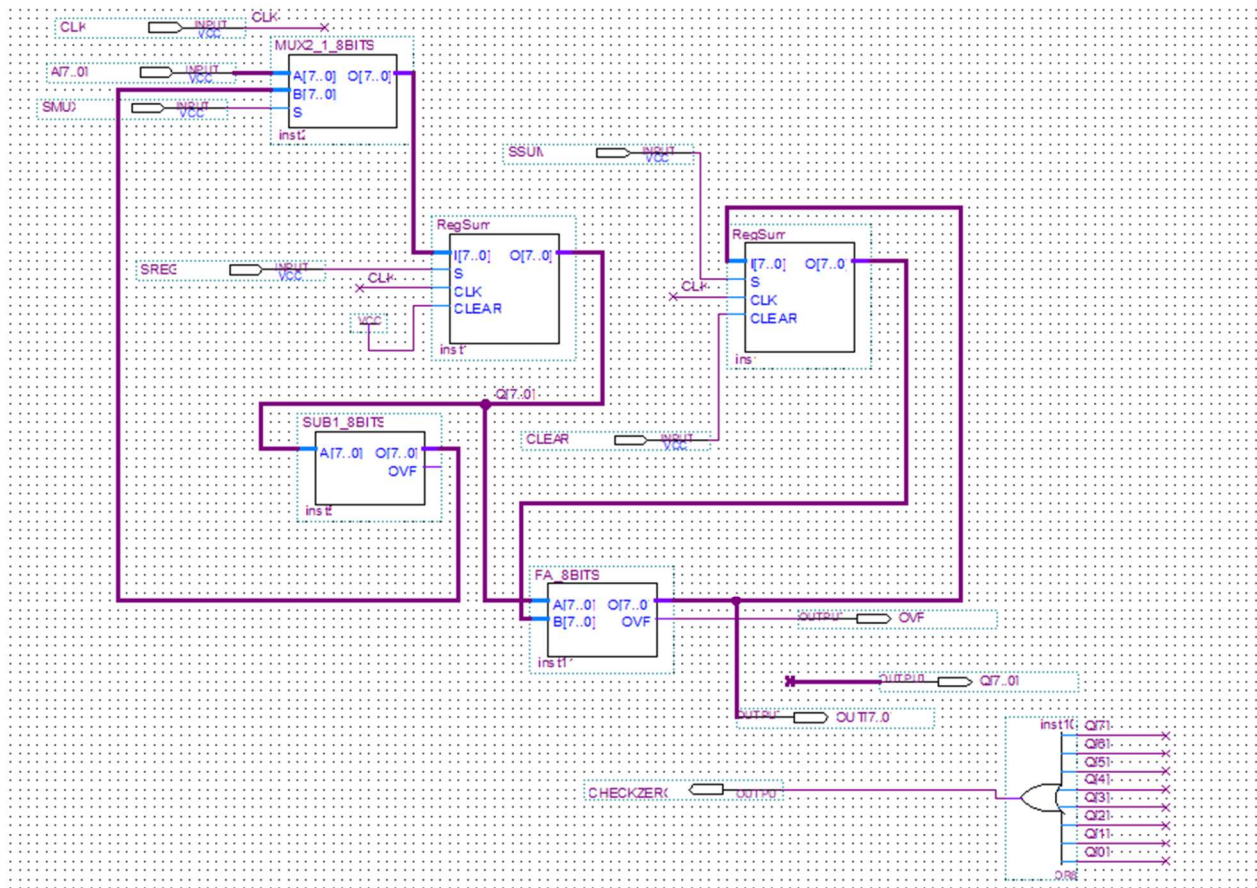
N—

End loop:

Output=RegSum

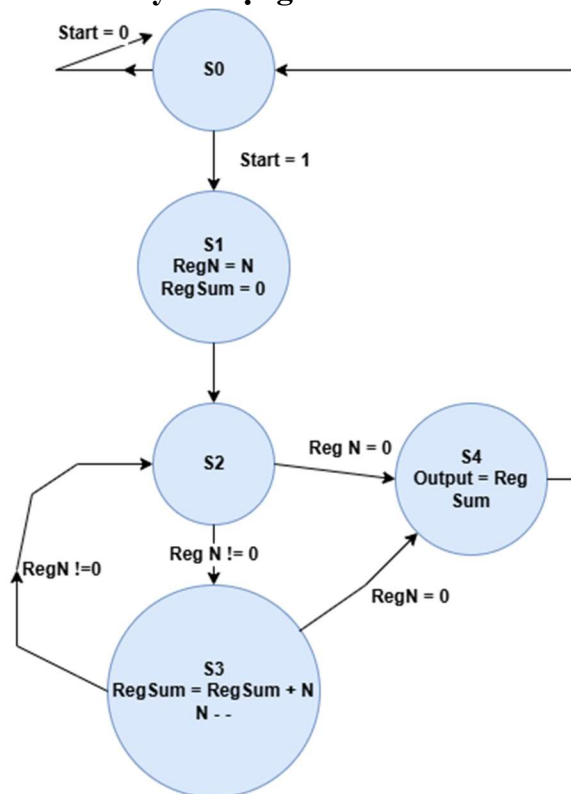
Sơ đồ datapath:

Gồm 2 thanh ghi để lưu giữ N và Sum, 2 khối tính toán tương tự như 1 ALU thực hiện phép $N -$ và $Sum = Sum + N$, 1 ngõ ra OR 8 bits để check_zero khi nào N trở về 0 dành cho khối control.



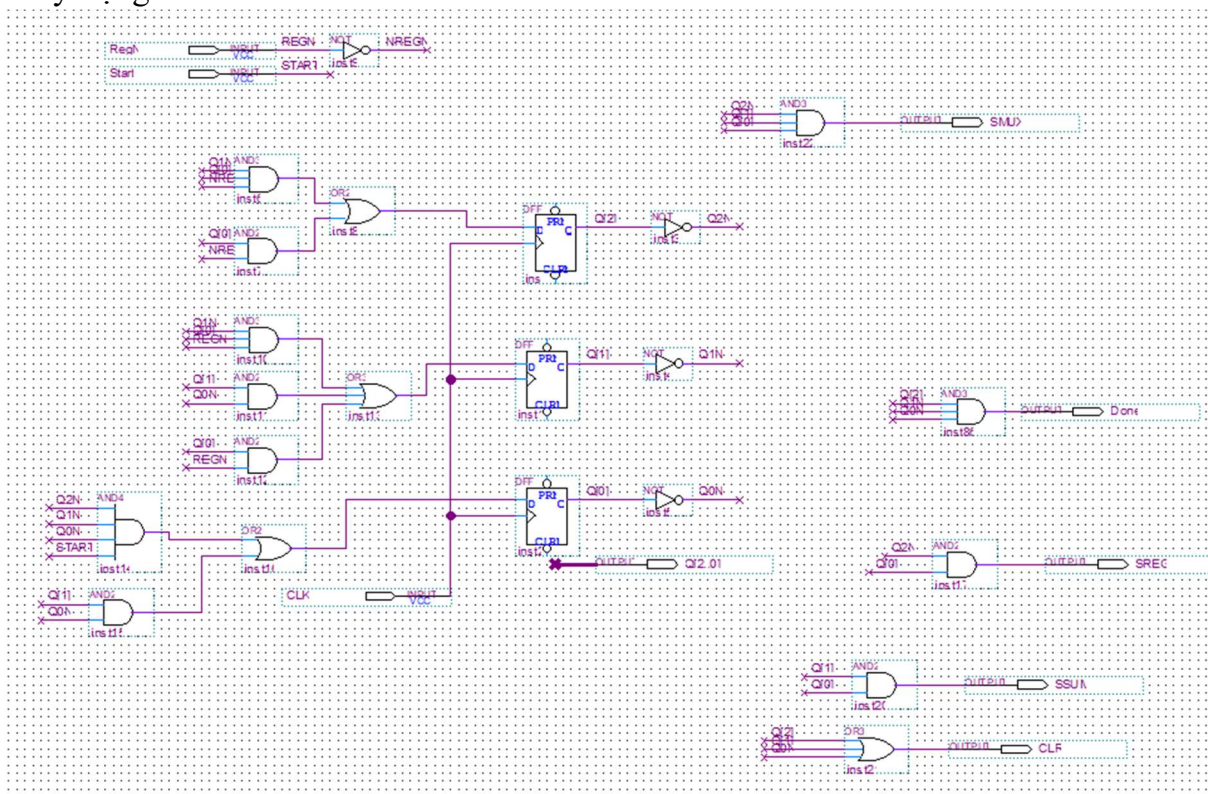
Ở đây dùng 2 khối riêng Sub1_8bits và FullAdder_8bits để giảm số lượng trạng thái và cho việc $\text{RegSum} = \text{RegSum} + N$ và $N \rightarrow$ thực hiện trong cùng 1 state.

Sơ đồ chuyển trạng thái của Control Unit:



A	B	C	D	E	F	G
Present State				Next State		Datapath Action
Q2	Q1	Q0	NAME	CONDITION	STATE	Operation
0	0	0	S0	Start=0	S0	DONE=0
				Start=1	S1	
0	0	1	S1		S2	RegN=import N
						RegSum=0
0	1	0	S2	RegN=0	S4	
				RegN!=0	S3	
0	1	1	S3	RegN=0	S4	RegSum = RegSum+N
				RegN!=0	S2	N--
1	0	0	S4		S0	DONE=1
						OutPort=RegSum

Máy trạng thái cho Control Unit:



$$D2=Q2_{next}= D2' D1 REGN$$

$$D1=Q1_{next}= D2' D1 REGN + D1' D0$$

$$D0=Q0_{next}= D2' D1' D0' START + D1 D0' REGN$$

Ngõ ra của Control Unit:

State	SMUX	SREG	SSUM	CLR	OE
S0	X	0	0	1	0
S1	0	1	0	0	0
S2	X	0	0	1	0
S3	1	1	1	1	0
S4	X	0	0	1	1

$SMUX = Q2' Q1 Q0$

$SREG = Q1' Q0 + Q1 Q0$

$SSUM = Q1 Q0$

$CLR = Q2 + Q1 + Q0'$

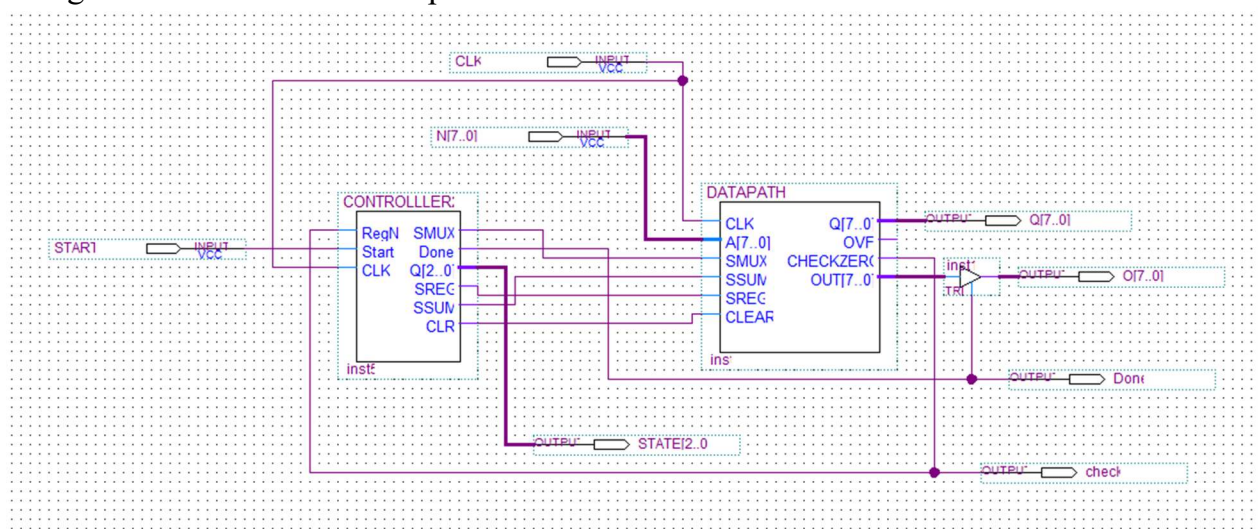
$DONE = Q2 Q1' Q0'$

Ngõ vào khối Control Unit là kiểm tra số N có bằng 0 hay chưa và tín hiệu Start:

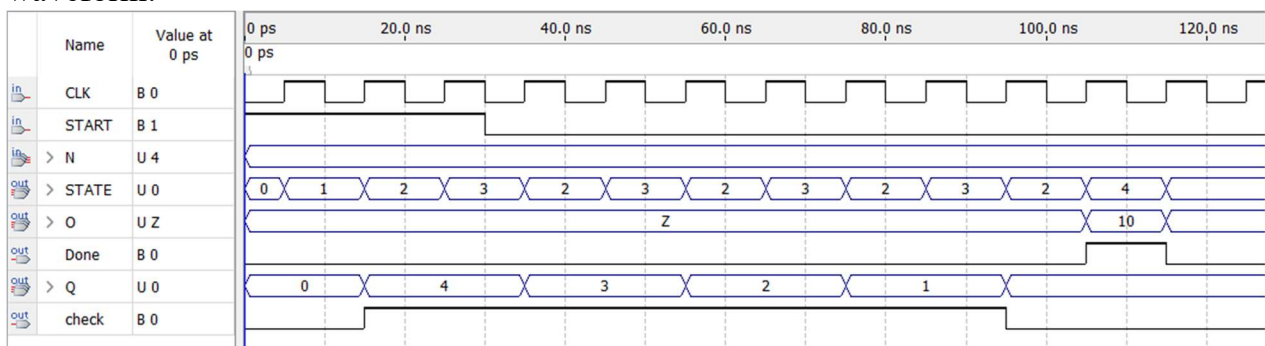
$RegN = Q7 + Q6 + Q5 + \dots + Q0$

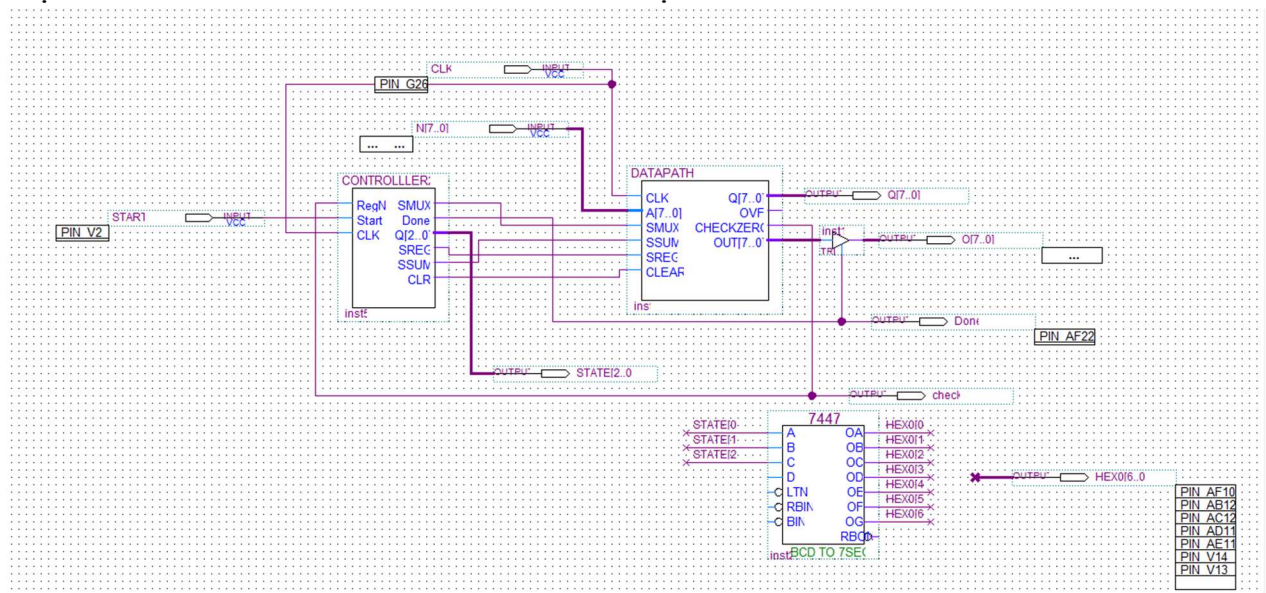
Start = Input

Nối giữa Control Unit và Datapath:



waveform:





Video nạp KIT với N=4: [video](#)