ĐỒ ÁN SOC:  
Xây dựng hệ thống SoC sử module Vector multiplier để thực hiện nhân 2 vector

1. Thiết kế mô đun:

Mô đun dùng nhân vô hướng 2 vecto.

Mô đun: multiple

Diagram

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên tín hiệu | Độ rộng (bit) | Hướng | Mô tả |
|  |  |  |  |  |
| 1 | iClk | 1 | Input | Cấp xung clock cho mô đun hoạt động |
| 2 | iReset | 1 | Input | Cấp tín hiệu reset mức thấp cho mô đun. |
| 3 | iChipSelect\_n | 1 | Input | Nếu iChipSelect\_n = 0, thì mô đun được phép hoạt động. Ngược lại, iChipSelect\_n = 1, mô đun không được hoạt động. |
| 4 | iWrite\_n | 1 | Input | Nếu iWrite\_n = 0, mô được cho phép dữ liệu vào. Ngược lại, oWrite\_n = 1, mô đun bỏ qua tín hiệu đưa vào. |
| 5 | iRead\_n | 1 | Input | Nếu iRead\_n = 0, mô đun cho phép đọc dữ liệu ra ngoài. Ngược lại, iRead\_n = 0, mô đun không cho phép đọc dữ liệu ra ngoài. |
| 6 | iAddress | 2 | Input | Địa chỉ cần cho việc đọc hoặc ghi. |
| 7 | iData | 32 | Input | Dữ liệu ghi vào. |
| 8 | oData | 32 | Output | Dữ liệu đọc ra ngoài. |

Các thanh ghi trong mô đun Multiple:

|  |  |  |  |
| --- | --- | --- | --- |
| Offset | Tên thanh ghi | Đọc/Ghi | Mô tả |
| 0 | a | Đọc/ghi | 32 bit dữ liệu giá trị a |
| 1 | b | Đọc/ghi | 32 bit dữ liệu giá trị a |
| 2 | c | Đọc/ghi | 32 bit dữ liệu giá trị a |
| 3 | d | Đọc/ghi | 32 bit dữ liệu giá trị a |
| 4 | control | Đọc/ghi | 4 bit dữ liệu cấu hình hoạt động cho module adder |

Diagram

Description automatically generatedMô đun adder:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên tín hiệu | Độ rộng (bit) | Hướng | Mô tả |
|  |  |  |  |  |
| 1 | iClk | 1 | Input | Cấp xung clock cho mô đun hoạt động |
| 2 | control | 4 | Input | [3:2:1:0] bit 3 =1 cho phép mô đun adder hoạt động, bit 2 sort reset tích cực mức thấp, bit 1,0 xét độ dài dữ liệu 00: độ rộng dữ liệu vào tối đa 4 bit, 01: 8 bit,11: 15 bit |
| 3 | a | 32 | Input | Dữ liệu vào |
| 4 | b | 32 | Input | Dữ liệu vào |
| 5 | c | 32 | Input | Dữ liệu vào |
| 6 | d | 32 | Input | Dữ liệu vào |
| 7 | status | 2 | Output | Trạng thái module 00: ko hoạt động, 01: đang tính toán,11 đã hoàn thành |
| 8 | ketqua | 32 | Output | Dữ liệu kết quả đọc ra ngoài. |

1. Tổng quan hệ thống

Diagram

Description automatically generatedHệ thống được thiết kế như hình bên dưới bên dưới:

1. Các bước tiến hành

• Tạo project Quartus tên là “soc”. Lưu ý đường dẫn thư mục project không được

có khoảng trắng.

• Chọn Family là Cyclone IV E, device là EP4CE115F29C7 nếu dùng board

DE2-115.

• Chọn Family là Cyclone V, device là 5CSXFC6D6F31C6 nếu dùng board

DE10-Standard.

o Trong project, tạo file mul.v và adder.v với nội dung được mô tả như 2

đoạn code bên dưới:

**module** mul**(**

**input** iClk**,**

**input** iReset\_n**,**

**input** iChipSelect\_n**,**

**input** iWrite\_n**,**

**input** iRead\_n**,**

**input** **[**3**:**0**]** iAddress**,**

**input** **[**31**:**0**]** iData**,**

**output** **reg** **[**31**:**0**]** oData

**);**

**reg** **[**31**:**0**]**a **,** b**,** c**,** d**;**

**reg** **[**3**:**0**]** control**;**

**wire** **[**31**:**0**]** kt**;**

**wire** **[**1**:**0**]** status**;**

**always@(posedge** iClk**,** **negedge** iReset\_n**)**

**begin**

**if(~**iReset\_n**)**

**begin**

oData **<=** 32'd0**;**

a **<=** 32'd0**;**

b **<=** 32'd0**;**

c **<=** 32'd0**;**

d **<=** 32'd0**;**

control**<=**4'd0**;**

**end**

**else**

**begin**

**if(** **~**iChipSelect\_n **&** **~**iWrite\_n**)**

**begin**

**case** **(**iAddress**)**

4'd0**:** a**[**31**:**0**]** **<=** iData**[**31**:**0**];**

4'd1**:** b**[**31**:**0**]** **<=** iData**[**31**:**0**];**

4'd2**:** c**[**31**:**0**]** **<=** iData**[**31**:**0**];**

4'd3**:** d**[**31**:**0**]** **<=** iData**[**31**:**0**];**

4'd4**:** control**[**3**:**0**]<=** iData**[**3**:**0**];**

**endcase**

**end**

**if** **(~**iChipSelect\_n **&** **~**iRead\_n**)**

**begin**

**case** **(**iAddress**)**

4'd0**:** oData **<=** a**;**

4'd1**:** oData **<=** b**;**

4'd2**:** oData **<=** c**;**

4'd3**:** oData **<=** d**;**

4'd4**:** oData**[**3**:**0**]** **<=** control**[**3**:**0**];**

4'd5**:** oData **<=** kt**;**

4'd6**:** oData**[**1**:**0**]** **<=** status**[**1**:**0**];**

**endcase**

**end**

**end**

**end**

adder adder1**(**

**.**clk**(**iClk**),**

**.**control**(**control**),**

**.**a**(**a**),**

**.**b**(**b**),**

**.**c**(**c**),**

**.**d**(**d**),**

**.**status**(**status**),**

**.**ketqua**(**kt**)**

**);**

**endmodule**

**module** adder**(**

**input** clk**,**

**input** **[**3**:**0**]** control**,**

**input** **[**31**:**0**]**a**,**

**input** **[**31**:**0**]**b**,**

**input** **[**31**:**0**]**c**,**

**input** **[**31**:**0**]**d**,**

**output** **reg** **[**1**:**0**]** status**,**

**output** **reg** **[**31**:**0**]** ketqua

**);**

**always@(posedge** clk**,** **negedge** control**[**2**])**

**begin**

**if(~**control**[**2**])**

**begin**

ketqua**<=** 32'h0**;**

status**<=** 2'h0**;**

**end**

**else**

**begin**

**if** **(**control**[**3**])**

**begin**

**if(**control**[**1**:**0**]==**3'b00**)**

**begin**

//status <= 2'b10;

ketqua **<=** a**[**3**:**0**]\***c**[**3**:**0**]+**b**[**3**:**0**]\***d**[**3**:**0**];**

status **<=** 2'b11**;**

**end**

**else** **if** **(**control**[**2**:**0**]==**3'b01**)**

**begin**

//status <= 2'b10;

ketqua **<=** a**[**7**:**0**]\***c**[**7**:**0**]+**b**[**7**:**0**]\***d**[**7**:**0**];**

status **<=** 2'b11**;**

**end**

**else** **if** **(**control**[**2**:**0**]==**3'b11**)**

**begin**

status **<=** 2'b10**;**

ketqua **<=** a**[**15**:**0**]\***c**[**15**:**0**]+**b**[**15**:**0**]\***d**[**15**:**0**];**

status **<=** 2'b11**;**

**end**

**end**

**else**

**begin**

status **<=** 2'b00**;**

ketqua **<=** 32'h0**;**

**end**

**end**

**end**

**endmodule**