明德扬
科技.教育

点拨·FPGA之用加法器代替乘法器并实现流水练习思路

点透学习误区　拨出设计精髓

主 讲：潘文明

明德扬科教

明德扬
科技·教育

QQ群： 97925396

官　网：http://www.mdy-edu.com

淘　宝：http://mdy-edu.taobao.com

课程大纲

1. 功能要求
2. 设计思路
3. 代码设计

一、功能要求

1、 内容：流水线设计的方法实现 4 输入的乘法运算；

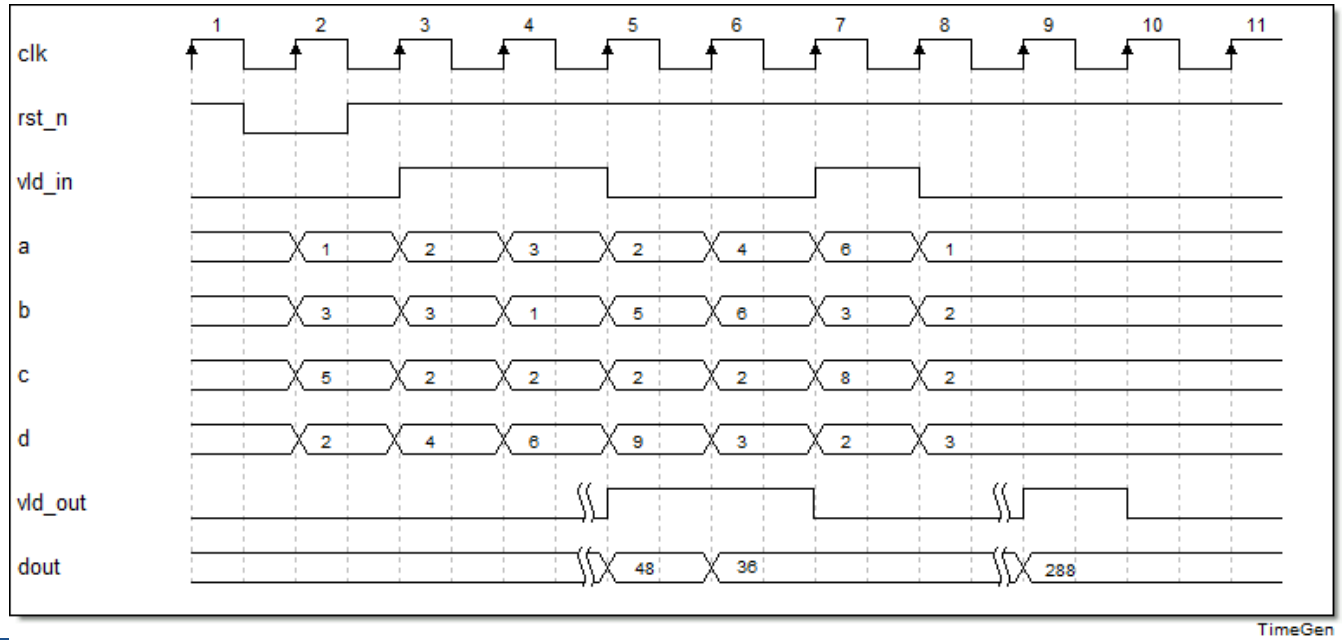
2、 要求：

a 不能用乘法器，要用加法器和选择器来代替；

b 寄存器之间至多只(一个加法器+一个选择器)；

c 流水线级数不限定；

d 要求能连续运算。



二、设计思路

1. 设计时要从输出信号倒推，现有条件不能直接得到的信号，先假设信号再设计；

2. 一个一个信号设计，逐个击破

三、设计思路—如何代替乘法器？

1. 如何用加法器和选择器代替乘法器？

2. 回顾下小学时乘法是如何一步一步计算，例如114*204（十进制），1101*1001（二进制）

```
   114
 X 204
-----
   456
  000
 228
-----
 23256
```

```
    1101
  X 1001
------
    1101
   0000
  0000
 1101
-------
 1110101
```

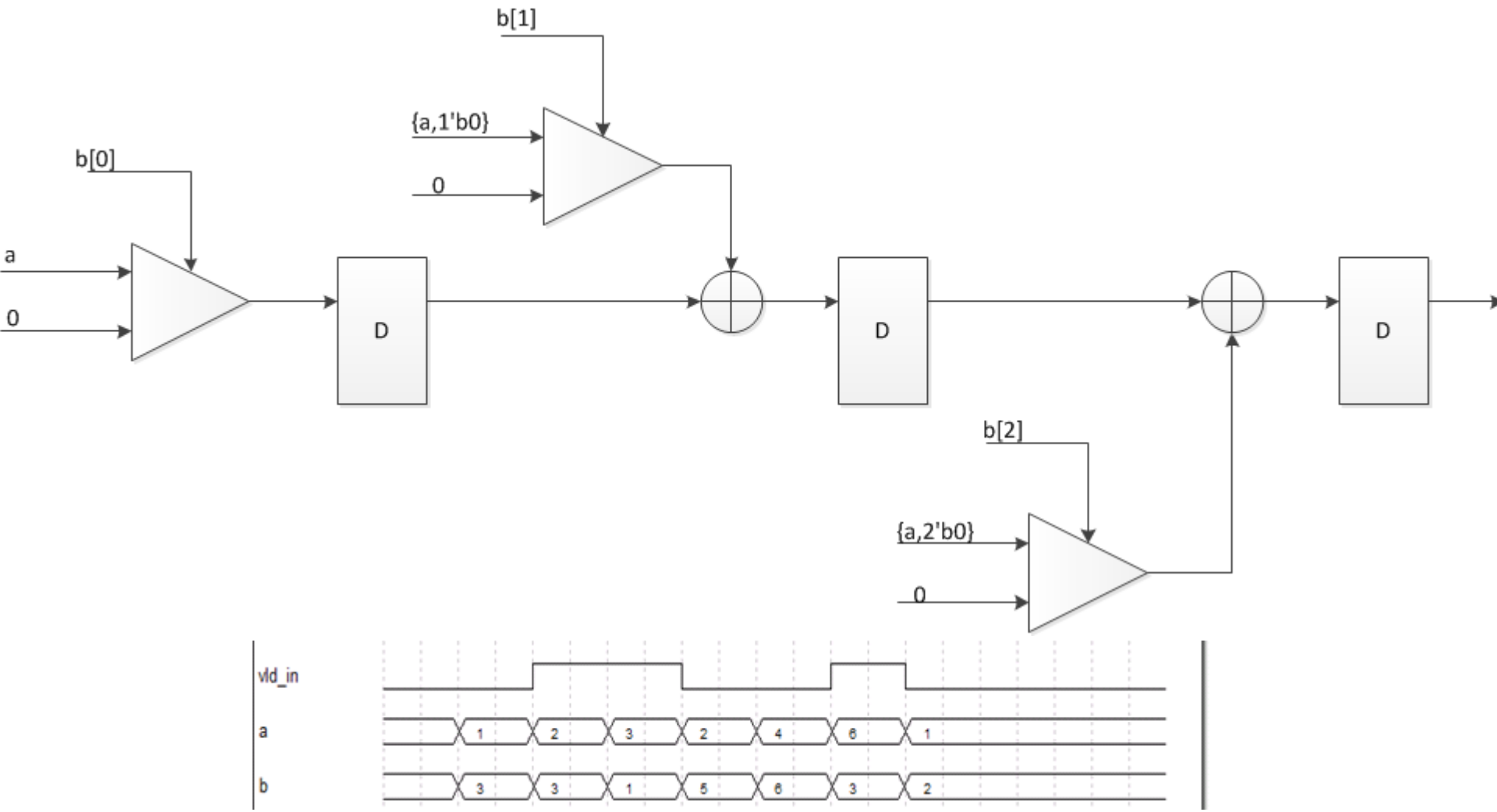
三、设计思路

1. a * b = a * b[0] + (a<<1)*b[1] + (a<<2)*b[2] + (a<<3)*b[3]

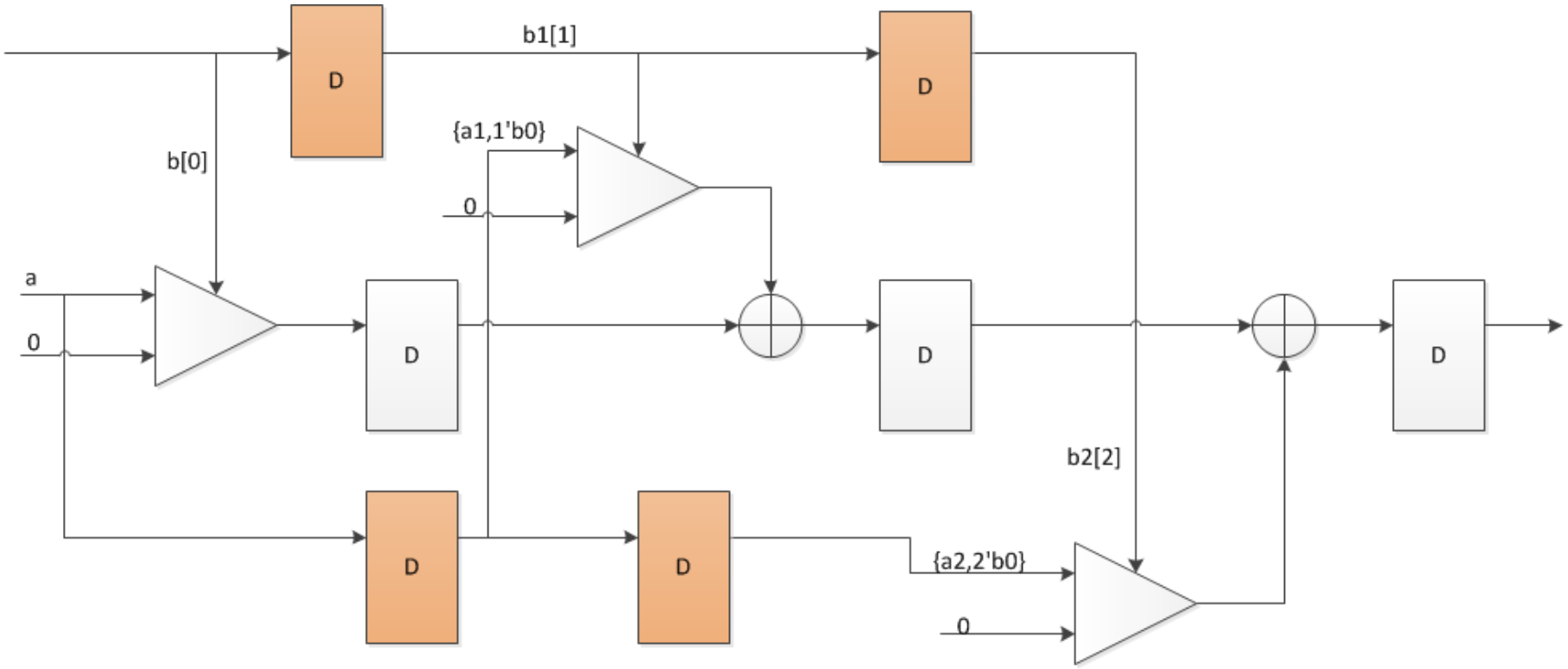
2. a*b[0] + {a,1’b0}*b[1] +{a,2’b0}*b[2] + {a,3’b0}*b[3]

3. (b[0]?a :0) + (b[1]?{a,1’b0} :0) + (b[2]?{a,2’b0} :0) + (b[3]?{a,3’b0} :0)

三、设计思路



三、设计思路



三、设计思路—vld_out

1. 请根据从输入数据到最后得到结构，需要多少拍，从而得到vld_out

三、设计思路—1个乘法器代码

1. 缓存数据

```
always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)begin
        a1 <= 0;
        a2 <= 0;
        a3 <= 0;
    end
    else begin
        a1 <= a;
        a2 <= a1;
        a3 <= a2;
    end
end

always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)begin
        b1 <= 0;
        b2 <= 0;
        b3 <= 0;
    end
    else begin
        b1 <= b[3:1];
        b2 <= b1[2:1];
        b3 <= b2[1];
    end
end
```

三、设计思路—1个乘法器代码

第一和第二个步骤

```
always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)begin
        sum1 <= 0;
    end
    else if(b[0]) begin
        sum1 <= a;
    end
    else begin
        sum1 <= 0;
    end
end

always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)begin
        sum2 <= 0;
    end
    else begin
        sum2 <= sum1 + ((b1[0])?({a1,1'b0}):0);
    end
end
```

三、设计思路—1个乘法器代码

第三和第四个结果

```
always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)begin
        sum3 <= 0;
    end
    else begin
        sum3 <= sum2 + ((b2[0])?({a2,2'b0}):0);
    end
end

always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)begin
        sum4 <= 0;
    end
    else begin
        sum4 <= sum3 + ((b3[0])?({a3,3'b0}):0);
    end
end
```

三、设计思路

1. 按同样的思路，补充完剩余代码

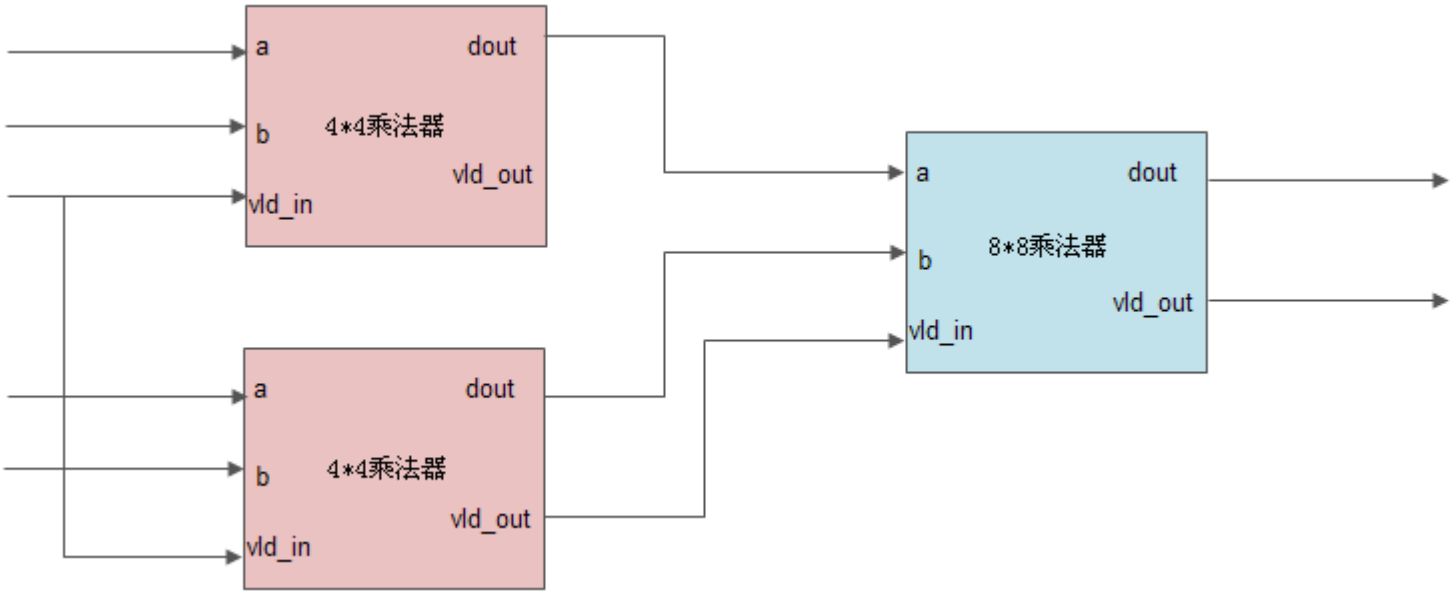
2. 模块复用的思想

三、设计思路—模块复用

1. 本练习需要用到3个乘法器，其中两个是4*4的乘法器，另一个是8*8的乘法器

2. 4*4的乘法器是相同的，因此要想到提高效率

3. 将前面的4*4乘法器设计为一个module，则可以例化来实现



三、设计思路—模块复用

1. 本练习需要用到3个乘法器，其中两个是4*4的乘法器，另一个是8*8的乘法器

2. 4*4的乘法器是相同的，因此要想到提高效率

3. 将前面的4*4乘法器设计为一个module，则可以例化来实现

4. Module的划分原则：功能明确，通过例化复用能提高效率。因为不宜过大或过小。

5. 4*4乘法器和8*8乘法器功能相同，能不能设计为一个module呢？答案当然是可以！通过参数例化来实现。（此处较复杂，先不详细讲。有兴趣的同学可以先研究，提示：用generate的方法）

明德扬科教

明德扬
科技·教育

QQ群： 97925396

官　网：http://www.mdy-edu.com

淘　宝：http://mdy-edu.taobao.com

Thank You !