

FPGA 试验心得

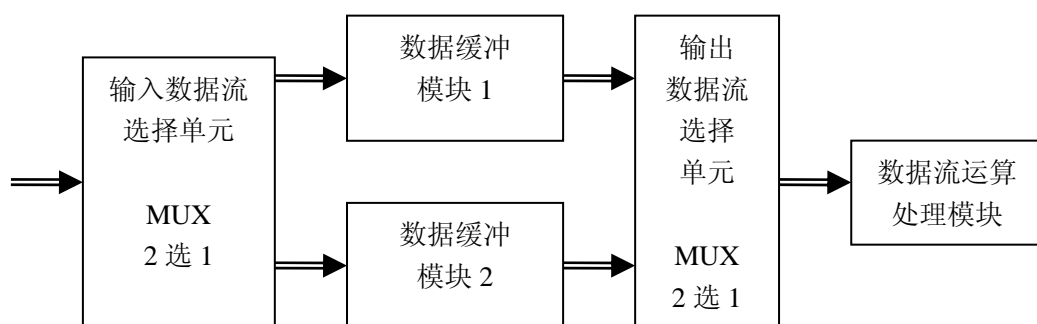
陈建军

2007-4-18

初步涉及到可编程，是在大学期间的 EDA 课上，当时只是将简单的 VHDL 语言程序烧录到芯片中去，从而来验证可以达到某种基本的功能。工作后，起初的 EMC 测试中，FPGA 的时钟信号线是一个比较严重的辐射源头。因此，在一般的印制板走线设计上，都会在时钟线上串联一个电阻，以期能够达到阻抗匹配的目的，从而避免谐振导致多次谐波，造成极高的辐射出来。同时，在 FPGA 芯片下方会在电源脚和信号地之间并联多个电容，滤波，用以保证在信号错综复杂的印制板上，达到理想的效果，避免干扰串进信号导致错误发生，引起时序或者 FPGA 错误。

经过 FPGA 这段时间的学习以及实践中，乒乓操作、串并转换、流水线操作、数据接口同步化，是逻辑设计的内在规律的体现，合理地采用这些设计思想能在 FPGA 设计工作中取得事半功倍的效果。

对于在刚刚接触 FPGA 的我而言，乒乓操作还是比较容易上手的，在基于 FPGA 的 PCI 设计中就有所应用。“乒乓操作”是一个常常被应用于数据流控制的处理技巧，类似于软件里面的队列。输入数据流通过“输入数据选择单元”将数据流等时分配到两个数据缓冲区，数据缓冲模块可以为任何存储模块，比较常用的存储单元为双口 RAM(DPRAM)、单口 RAM(SPRAM)、FIFO 等。

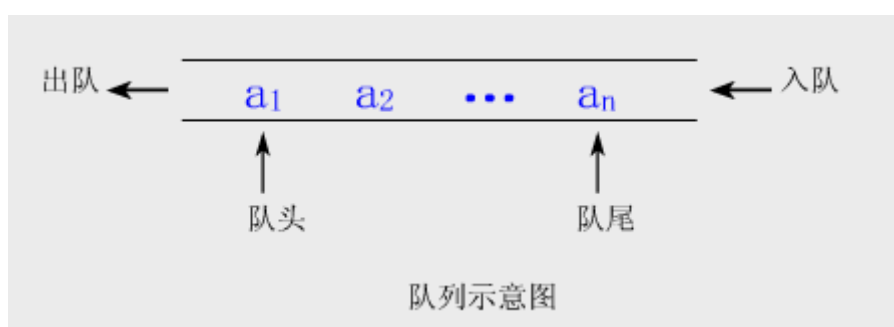


乒乓操作图示

在第一个缓冲周期，将输入的数据流缓存到“数据缓冲模块 1”；在第 2 个缓冲周期，通过“输入数据选择单元”的切换，将输入的数据流缓存到“数据缓冲模块 2”，同时将“数据缓冲模块 1”缓存的第 1 个周期数据通过“输入数据选择单元”的选择，送到“数据流运算处理模块”进行运算处理；在第 3 个缓冲周期通过“输入数据选择单元”的再次切换，将输入的数据流缓存到“数据缓冲模块 1”，同时将“数据缓冲模块 2”缓存的第 2 个周期的数据通过“输入数据选择单元”切换，送到“数据流运算处理模块”进行运算处理。如此循环。

乒乓操作的最大特点是通过“输入数据选择单元”和“输出数据选择单元”按节拍、相互配合的切换，将经过缓冲的数据流没有停顿地送到“数据流运算处理模块”进行运算与处理。把乒乓操作模块当做一个整体，站在这个模块的两端看数据，输入数据流和输出数据流都是连续不断的，没有任何停顿，因此非常适合对数据流进行流水线式处理。所以乒乓操作常常应用于流水线式算法，完成数据的无缝缓冲与处理。乒乓操作还具有的优点有可以节约缓冲空间，同时还可以达到低速模块处理高速数据流的效果。通过乒乓操作实现低速模块处理高速数据的实质是：通过 DPRAM 这种缓存单元实现了数据流的串并转换，并行用“数据预处理模块 1”和“数据预处理模块 2”处理分流的数据，是面积与速度互换原则的体现！

常见的乒乓操作是 FIFO, first in first out, 顾名思义，是一种可以实现数据先入先出的存储器件。FIFO 就像一个单向管道，数据只能按照固定的方向从管道一头进来，再按照相同的顺序从管道的另一头出去，最先进来的数据必定是最先出去。这类似于软件中的队列。



FIFO的基本单元是寄存器，作为存储器件，FIFO的存储能力是由其内部定义的存储寄存器的数量决定的。当然 a_n 的位宽可以按照处理数据的位宽来设定，如 8 位、16 位等，存储单元的个数，即 n 值，是由处理数据量来定，一个存储单元存储一个数据。在每一个时钟上升沿到来时，数据向左移动一个存储单元。这样在时钟的控制下，数据从左至右通过存储单元。

下面以建立一个 FIFO 的模型说明：举例一个 16x16 的 FIFO，内建 16 个存储单元，每

个存储单元可以存储一个 16 位的数，所以该 FIFO 每次都可以输入一个 16 位的数，输出也一样。

```
module
fifo(clock, reset, read, write, fifo_in, fifo_out, fifo_empty, fifo_half, fifo_full)
input clock, reset, read, write;
input [15:0] fifo_in;
output [15:0] fifo_out;
output fifo_empty, fifo_half, fifo_full;

reg [15:0] fifo_out;
reg [3:0] read_ptr, write_ptr, counter;
reg [15:0] ram [15:0];
wire fifo_empty, fifo_half, fifo_full;

always @(posedge clock)
if(reset)
begin
read_ptr=0;
write_ptr=0;
counter=0;
fifo_out=0;
end
else
case{(read, write)}
2' b00;
counter=counter;

2' b01;
Begin
ram[write_ptr]=fifo_in;
```

```

        counter=counter+1;
        write_ptr=(write_ptr==15)?0:write_ptr+1;
    end

2' b10:
begin
    fifo_out = ram[read_ptr1];
    counter=counter-1;
    read_ptr=(read_ptr==15)?0:read_ptr+1;
    (read_ptr==15)
end

2' b11:
begin
    if(counter==0)
        fifo_out=fifo_in;
    else
        begin
            ram[write_ptr]=fifo_in;
            fifo_out=ram[rear_ptr];
            write_ptr=( Write_ptr==15)?0:write_ptr+1;
            read_ptr=(read_ptr==15)?0:read_ptr+1;
        end
    end
endcase
assign fifo_empty=(counter==0)
assign fifo_full=(counter==15)
assign fifo_half=(counter==8)

endmodule

```

对于包括 FIFO 在内的乒乓操作方式，在 FPGA 设计中还是比较实用的，工作原理简单，因此易于操作控制。其最大特点是通过“输入数据选择单元”和“输出数据选择单元”，按节拍、相互配合地切换，将经过缓冲的数据流不停顿地送到“数据流运算处理模块”，进行运算及处理。把乒乓操作模块当作一个整体，此模块两端的输入数据流与输出数据流均是连续不断的，没有任何停顿，因此非常适合进行流水线式处理，完成数据的无缝缓冲与处理。乒乓操作可应用于包括高速数据采集、多处理器接口和通信中的高速缓冲等各种领域。