

HDMI（二）：原语

目录

- 1、OSERDESE2（并转串）
- 2、OBUFDS（差分输出）

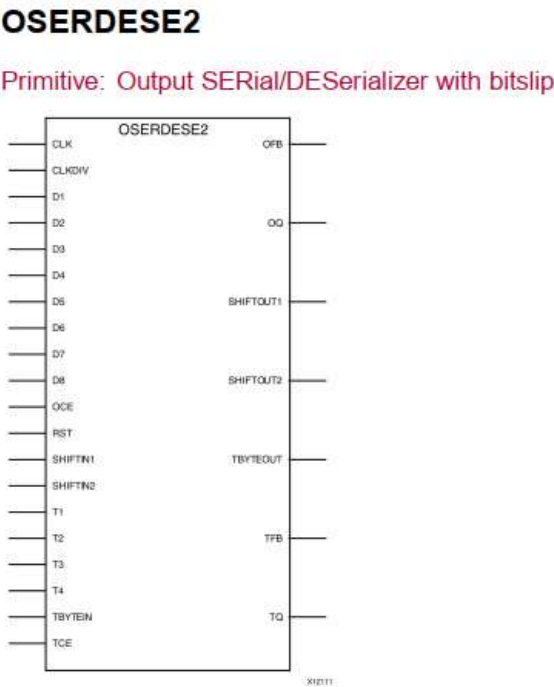
原语（primitive）：在操作系统中，一般是指由若干条指令组成的程序段，用来实现某个特定功能，在执行过程中不可被中断。而在硬件设计中，原语相当于就是软件中的库函数，是芯片设计厂家为了方便开发而设计的一种专用的高速的硬件电路设计。

我们可以使用这些原语来加速开发，比如说我们需要并串转换，我们可以自己写逻辑，这样虽然可以实现功能，但是速度和稳定性可能会稍差。如果用原语，因为原语是芯片厂家设计的，芯片厂家肯定比使用者更熟悉芯片的结构，所以原语相比自己设计逻辑，一般会有更好的资源利用率，更加高速稳定。

Xilinx 原语是 Xilinx 器件底层硬件中的功能模块，它使用专用的资源来实现一系列的功能。相比于 IP 核，原语的调用方法更简单，但是一般只用于实现一些简单的功能。

1、OSERDESE2（并转串）

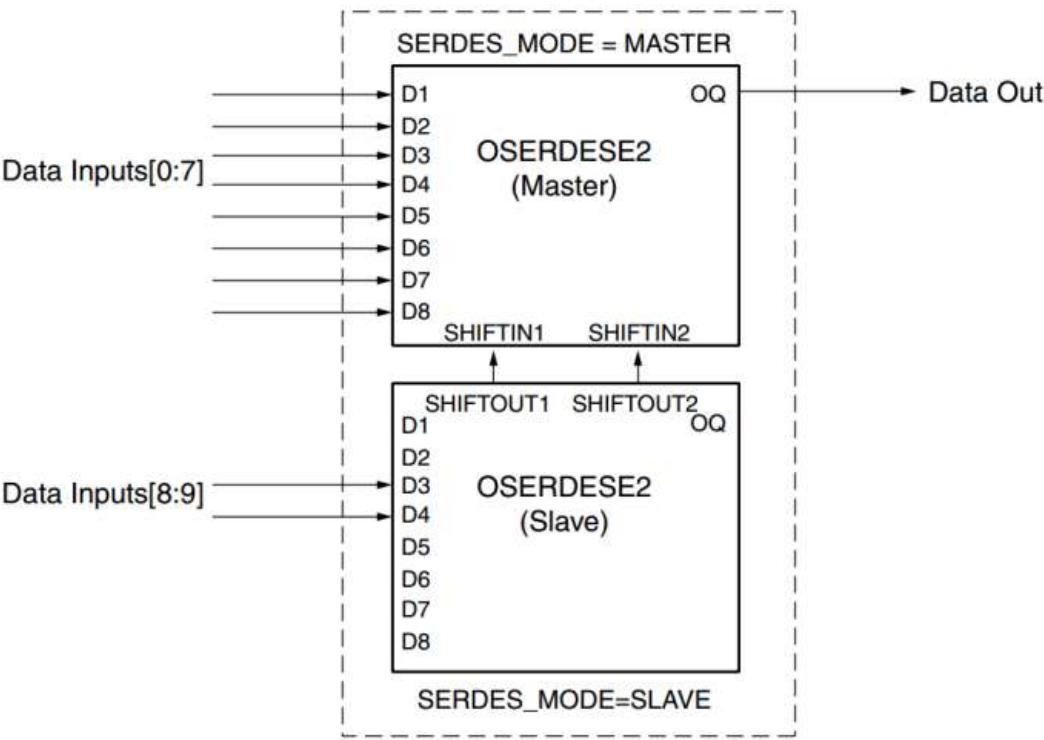
Xilinx 中的 OSERDESE2是一个最多8：1的并串转换原语



但是我们在设计HDMI时，需要进行10位转一位的并串转换，所以我们需要级联两个OSERDESE2原语来进行位扩展。

如果进行OSERDESE2级联，那么OSERDESE2就会分为主机（Master）和从机（Slave），在进行位扩展时，从机只能使用D3—D8

D1 - D8	Input	1	All incoming parallel data enters the OSERDES module through ports D1 to D8. These ports are connected to the FPGA fabric, and can be configured from two to eight bits (i.e., a 8:1 serialization). Bit widths greater than six (up to 14) can be supported by using a second OSERDES in SLAVE mode.
---------	-------	---	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



在Xilinx的UG768《Xilinx 7Series FPGA Libraries Guide for HDL Designs》中，有参考程序

其中的DDR是双倍数据速率的意思，可以在上升沿和下降沿进行操作，达到了速率翻倍的的效果。
10:1 的并转串过程所生成的串行数据速率是实际像素时钟速率的 10 倍。按理来说OSERDESE2应该输入10倍像素时钟的CLK，由于有DDR，所以只需要产生一个5倍像素时钟的CLK输入OSERDESE2

```
// OSERDESE2: Output SERIAL/DESerializer with bitslip
// 7 Series
// Xilinx HDL Libraries Guide, version 13.1

OSERDESE2 #(
    .DATA_RATE_OQ("DDR"), // "SDR" or "DDR"
    .DATA_RATE_TQ("DDR"), // "BUF", "SDR" or "DDR"
    .DATA_WIDTH(4), // Parallel data width (2-8,10)
    .INIT_OQ(1'b0), // Initial value of OQ output (0/1)
    .INIT_TQ(1'b0), // Initial value of TQ output (0/1)
    .SERDES_MODE("MASTER"), // "MASTER" or "SLAVE"
    .SRVAL_OQ(1'b0), // OQ output value when SR is used (0/1)
    .SRVAL_TQ(1'b0), // TQ output value when SR is used (0/1)
    .TBYTE_CTL("FALSE"), // Enable tristate byte operation ("TRUE" or "FALSE")
    .TBYTE_SRC("FALSE"), // Tristate byte source ("TRUE" or "FALSE")
    .TRISTATE_WIDTH(4) // 3-state converter width (1 or 4)
)
OSERDESE2_inst (
    .OQB(OQB), // 1-bit output: Feedback path for data output
    .OQ(OQ), // 1-bit output: Data path output
    // SHIFTOUT1/SHIFTOUT2: 1-bit (each) output: Data output expansion (1-bit each)
    .SHIFTOUT1(SHIFTOUT1),
    .SHIFTOUT2(SHIFTOUT2),
    .TBYTEOUT(TBYTEOUT), // 1-bit output: Byte group tristate output
    .TFB(TFB), // 1-bit output: 3-state control output
    .TQ(TQ), // 1-bit output: 3-state control output
    .CLK(CLK), // 1-bit input: High speed clock input
    .CLKDIV(CLKDIV), // 1-bit input: Divided clock input
    // D1 - D8: 1-bit (each) input: Parallel data inputs (1-bit each)
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .D5(D5),
    .D6(D6),
    .D7(D7),
    .D8(D8),
    .OCE(OCE), // 1-bit input: Output data clock enable input
    .RST(RST), // 1-bit input: Reset input
    // SHIFTIN1/SHIFTIN2: 1-bit (each) input: Data input expansion (1-bit each)
    .SHIFTIN1(SHIFTIN1),
    .SHIFTIN2(SHIFTIN2),
    // T1 - T4: 1-bit (each) input: Parallel 3-state inputs
    .T1(T1),
    .T2(T2),
    .T3(T3),
    .T4(T4),
    .TBYTEIN(TBYTEIN), // 1-bit input: Byte group tristate input
    .TCE(TCE) // 1-bit input: 3-state clock enable input
);
```

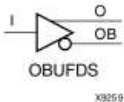
DDR (double data rate, 双倍数据速率) 在时钟的上升沿和下降沿采样

2、OBUFDS (差分输出)

这个原语比较简单，一个数据输入，两个差分输出。

OBUFDS

Primitive: Differential Signaling Output Buffer



Logic Table

Inputs	Outputs	
I	O	OB
0	0	1
1	1	0

Verilog Instantiation Template

```
// OBUFDS: Differential Output Buffer
// 7 Series
// Xilinx HDL Libraries Guide, version 13.1

OBUFDS #(
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFDS_inst (
    .O(O), // Diff_p output (connect directly to top-level port)
    .OB(OB), // Diff_n output (connect directly to top-level port)
    .I(I) // Buffer input
);

// End of OBUFDS_inst instantiation
```

其他更多的Xilinx原语可以参考UG768 《Xilinx 7Series FPGA Libraries Guide for HDL Designs》

博客的话，可以参考：<http://www.elecfans.com/pld/845058.html>

芯片设计入门课程（含链接）免费教程

芯片设计学习课程，包含数字IC，模拟IC等多个方向课程

广告