

مستندات کد Verilog شیفت‌دهنده 4 بیتی

مقدمه:

این کد Verilog یک شیفت‌دهنده 4 بیتی را پیاده‌سازی می‌کند که می‌تواند بیت‌های ورودی را به سمت چپ یا راست جابجا کند. این شیفت‌دهنده می‌تواند برای انجام انواع مختلف عملیات منطقی، مانند جمع و تفریق دودویی، ضرب و تقسیم دودویی و تبدیل اعداد به فرم‌های مختلف استفاده شود.

شرح کد:

1. ماژول:

Verilog

```
module shifter4bit
    ,input clk
    ,input rst
    ,input [3:0] data_in
    ,input sel
    شیفت (0: راست، 1: چپ)
    output [3:0] data_out
; (
```

این کد یک ماژول به نام shifter4bit تعریف می‌کند که دارای 5 پورت است:

• clk: ورودی کلاک

• rst: ورودی ریست

• data_in: داده ورودی 4 بیتی

• sel: ورودی انتخاب جهت شیفت (0: راست، 1: چپ)

• data_out: داده خروجی 4 بیتی

2. رجیسترها:

Verilog

```
// رجیسترها برای ذخیره داده‌ها  
;reg [3:0] reg1, reg2
```

این کد دو رجیستر 4 بیتی به نام reg1 و reg2 تعریف می‌کند. این رجیسترها برای ذخیره داده‌های ورودی و خروجی شیفت‌دهنده استفاده می‌شوند.

3. بلوک always:

Verilog

```
// بلوک always فعال با لبه مثبت کلاک  
always @(posedge clk) begin  
    if (rst) begin  
        // در صورت ریست فعال، رجیسترها را  
        // با 0 مقداردهی اولیه می‌کنیم  
        ;reg1 <= 0  
        ;reg2 <= 0  
    end else begin  
        // در غیر این صورت، داده‌ها را بر  
        // اساس ورودی انتخاب به رجیسترها منتقل می‌کنیم  
        ;reg1 <= data_in  
        if (sel) begin  
            // شیفت چپ  
            ;reg2 <= {reg1[2:0], 1'b0}  
        end else begin  
            // شیفت راست  
            ;reg2 <= {1'b0, reg1[3:1]}  
        end  
    end  
end
```

این کد یک بلوک always فعال با لبه مثبت کلاک تعریف می‌کند. این بلوک در هر لبه مثبت کلاک اجرا می‌شود.

در داخل بلوک always، دو شرط بررسی می‌شود:

• **شرط ریست:** اگر ورودی rst فعال باشد، رجیسترها با 0 مقداردهی اولیه می‌شوند.

• **شرط انتخاب جهت شیفت:** اگر ورودی sel 0 باشد، بیت‌های ورودی به سمت راست جابجا می‌شوند. اگر ورودی sel 1 باشد، بیت‌های ورودی به سمت چپ جابجا می‌شوند.

4. خروجی:

Verilog

```
// خروجی شیفت‌دهنده  
;assign data_out = reg2
```

این کد خروجی شیفت‌دهنده را به عنوان data_out تعیین می‌کند. خروجی شیفت‌دهنده برابر با مقدار رجیستر reg2 است.

نحوه استفاده از کد:

برای استفاده از این کد، باید آن را در یک فایل با نام دلخواه v ذخیره کنید و سپس از آن در یک پروژه Verilog استفاده کنید.

برای شبیه‌سازی کد، می‌توانید از یک شبیه‌ساز Verilog مانند Xilinx Vivado یا Mentor Graphics ModelSim استفاده کنید.

نکات:

- این کد از یک بلوک always فعال با لبه مثبت کلاک برای به‌روزرسانی رجیسترها در هر لبه مثبت کلاک استفاده می‌کند.
- از multiplexers برای انتخاب جهت شیفت (راست یا چپ) استفاده می‌شود.
- می‌توانید کد را با اضافه کردن ورودی‌های دیگر مانند ورودی "فعال‌سازی" و "سرعت شیفت" گسترش دهید.