模拟与数字电路

Analog and Digital Circuits

第 十 讲： **D型触发器的应用**

Lecture 10: **Applications of DFFs**

主　　讲： 陈 迟 晓

Instructor : Chixiao Chen

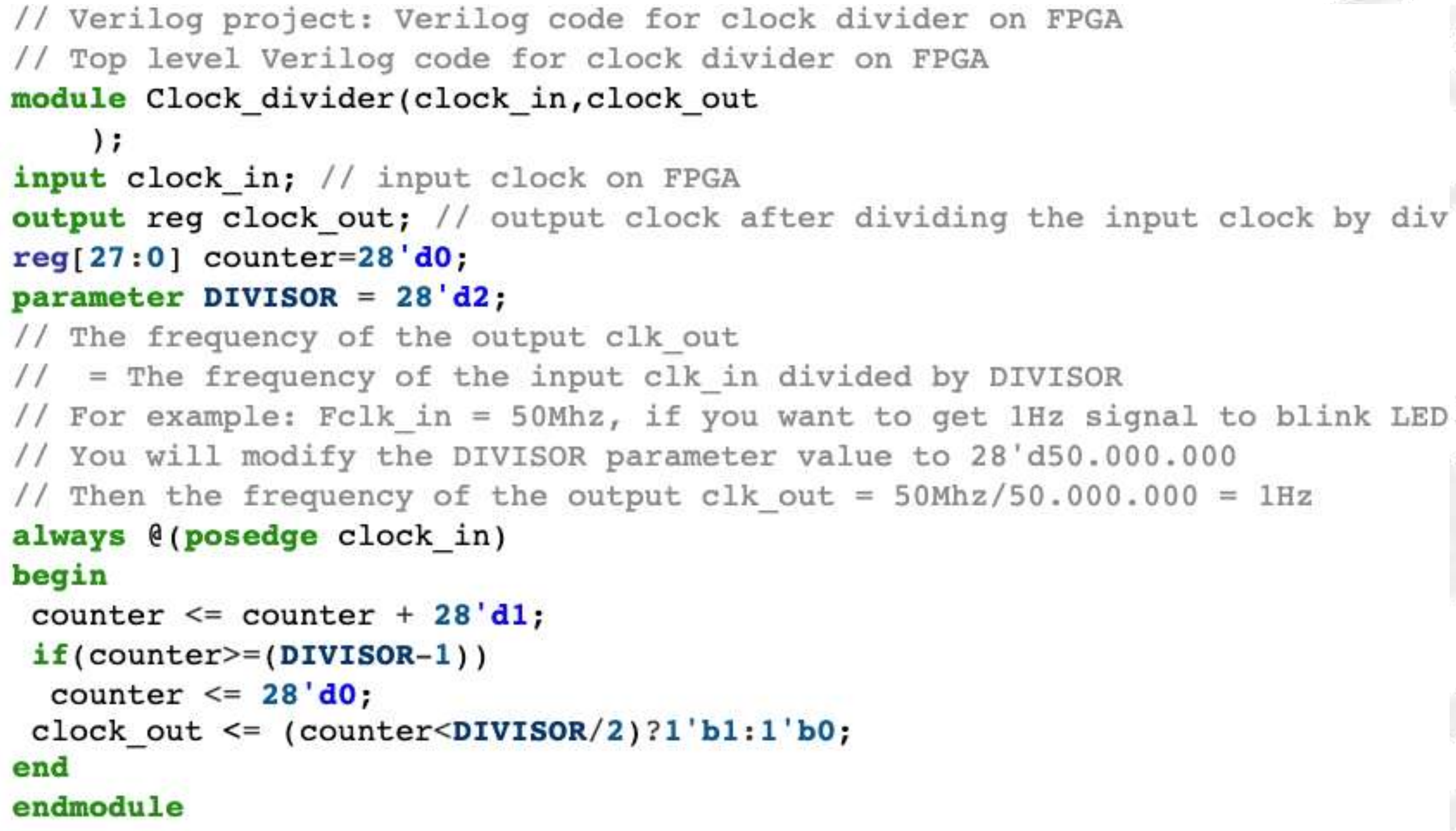
课程主页 扫一扫

提纲

- 复习
  - 请描述下页列电路的功能，并用verilog藐视

  - 分频器
  - 移位寄存器
  - 同步时序电路
  - 流水线电路

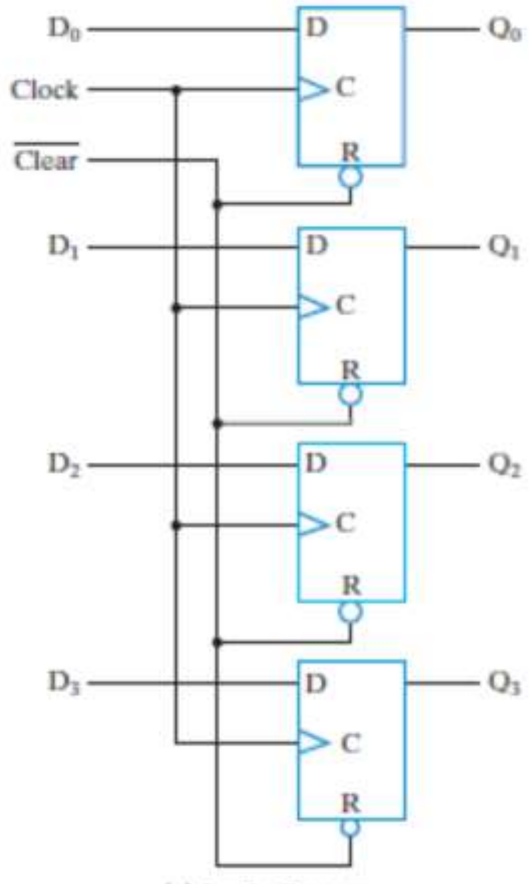
任意整数倍分频电路



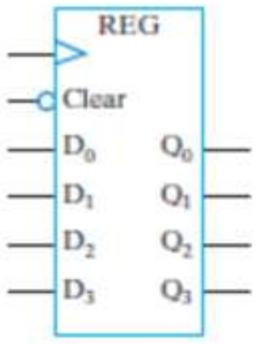
```
// Verilog project: Verilog code for clock divider on FPGA
// Top level Verilog code for clock divider on FPGA
module Clock_divider(clock_in,clock_out
    );
input clock_in; // input clock on FPGA
output reg clock_out; // output clock after dividing the input clock by div
reg[27:0] counter=28'd0;
parameter DIVISOR = 28'd2;
// The frequency of the output clk_out
//  = The frequency of the input clk_in divided by DIVISOR
// For example: Fclk_in = 50Mhz, if you want to get 1Hz signal to blink LED
// You will modify the DIVISOR parameter value to 28'd50.000.000
// Then the frequency of the output clk_out = 50Mhz/50.000.000 = 1Hz
always @(posedge clock_in)
begin
 counter <= counter + 28'd1;
 if(counter>=(DIVISOR-1))
  counter <= 28'd0;
 clock_out <= (counter<DIVISOR/2)?1'b1:1'b0;
end
endmodule
```

DFF应用(一) 寄存器 – Register

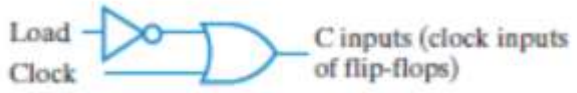
- Register 寄存器：用于实现数据存储与数据移动的操作
  - N-bit register store n-bit binary information
  - 带有载入使能（load enable）的门控寄存器



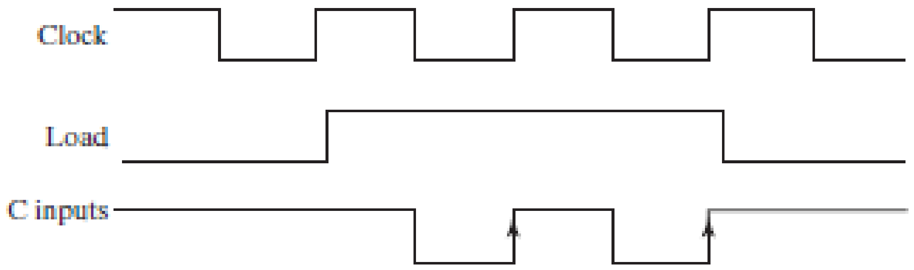
(a) Logic diagram



(b) Symbol



(c) Load control input

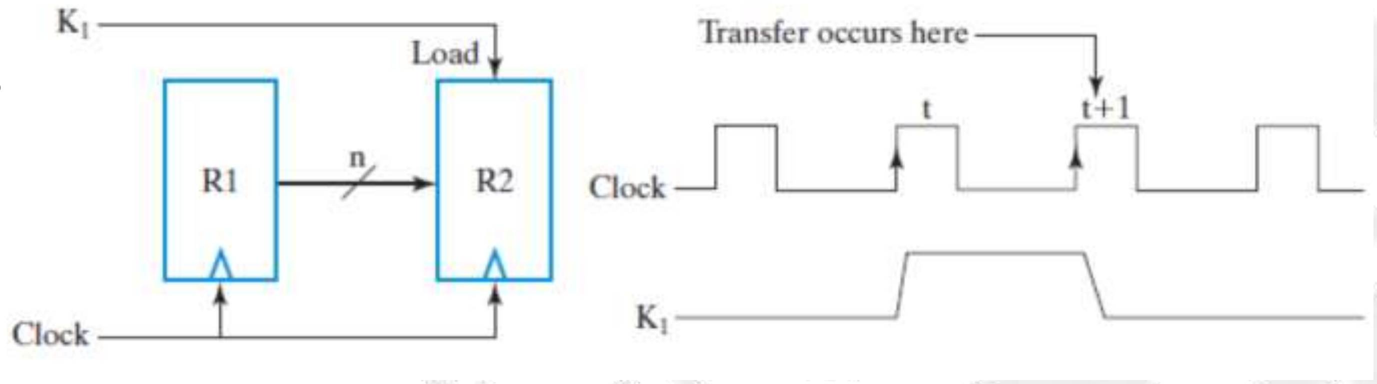


(d) Timing diagram

Register Transfer

- **大型数字系统中**，我们常用计算器转移级（RTL）来描述各个电路某块，因此我们把这样的Verilog代码简称为RTL代码

- Data transfer from a register to another one
  - R2 ←R1
- Data transfer with conditions
  - If (K1=1) then (R2 ←R1)
  - K1: R2 ←R1
- Hardware implementation



移位寄存器与串并转换

- 移位寄存器是在若干相同时间脉冲下工作的以触发器为基础的器件，数据以并行或串行的方式输入到该器件中，然后每个时间脉冲依次向左或右移动一个比特，在输出端进行输出。

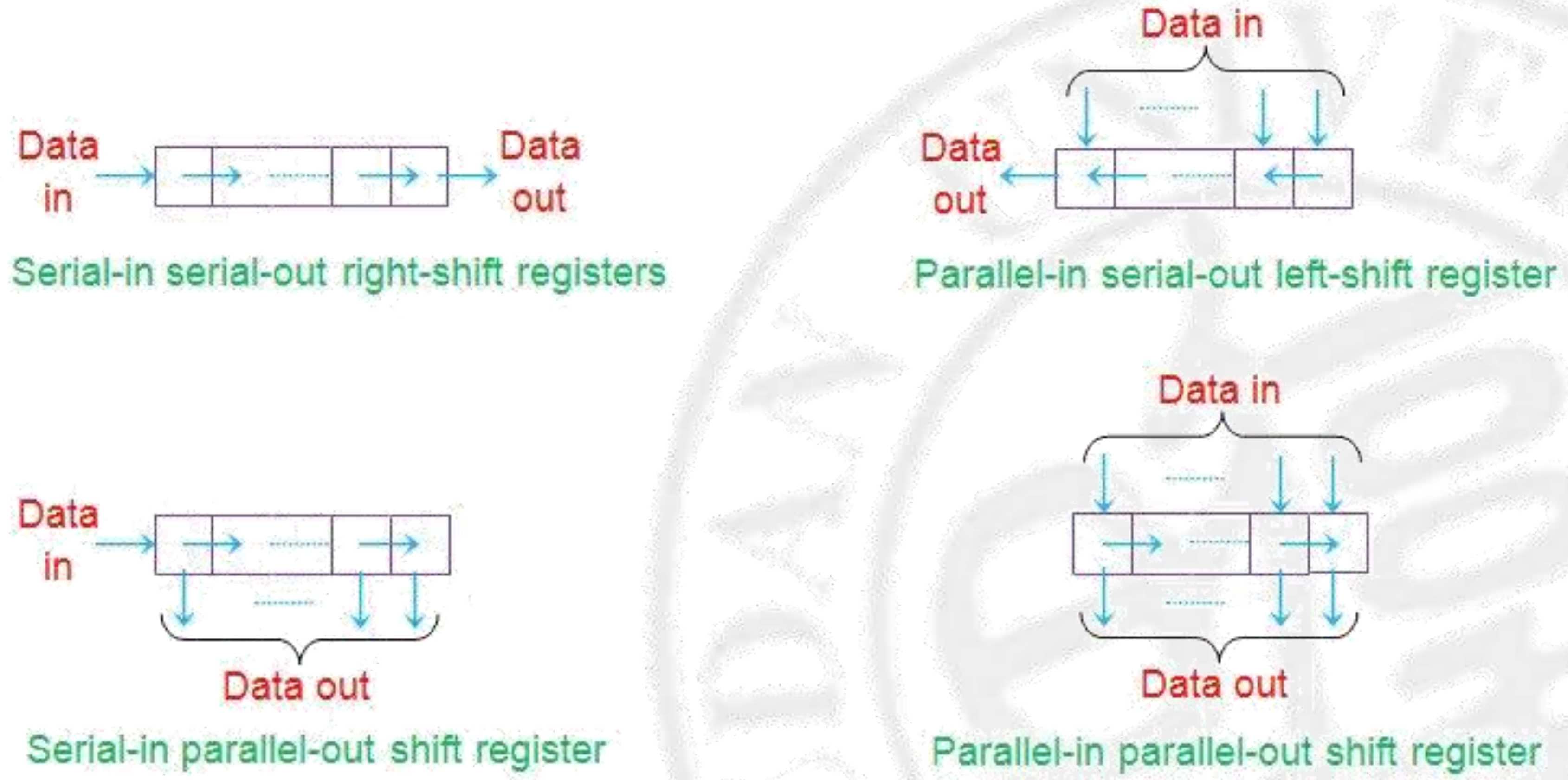

Data in
Data out
Serial-in serial-out right-shift registers
Data in
Data out
Parallel-in serial-out left-shift register
Data in
Data out
Serial-in parallel-out shift register
Data in
Data out
Parallel-in parallel-out shift register

Figure 2 Various kinds of shift registers

移位寄存器

- 具有同步预置功能的8位寄存器

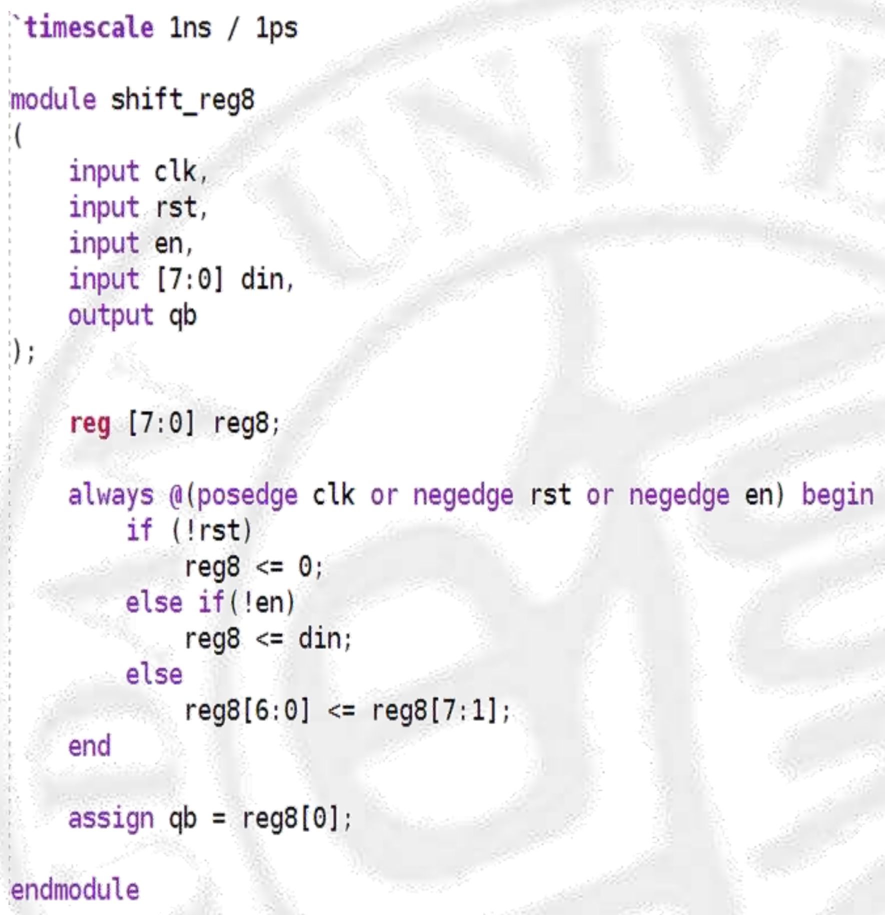
□ clk是移位时钟信号，load是并行数据预置使能信号，din是8位并行预置数据端口，qb是串行输出端口。

工作原理：

- 当clk的上升沿到来时，过程被启动，如果此时预置使能端口load为高电平，则输入端口din的8位二进制数被同步并行移入移位寄存器，用作串行右移的初始值；
- 如果此时预置使能端口load为低电平，则执行赋值语句：

  reg8[6:0] <= reg8[7:1];

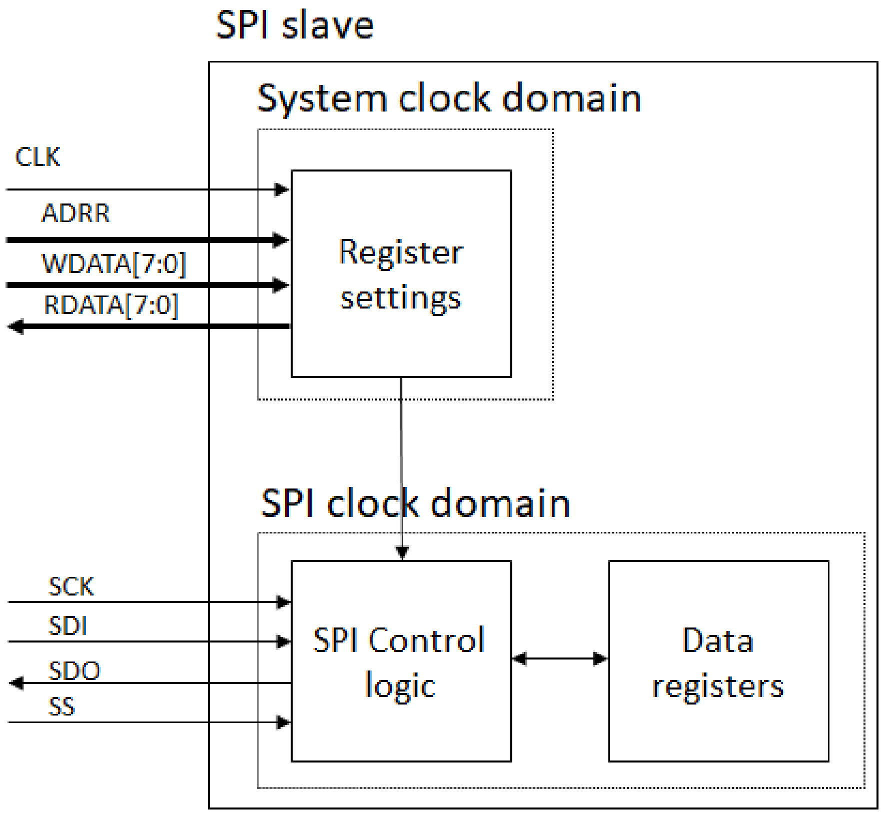
这样完成一个时钟周期后，把上一时钟周期的高7位值reg8[7:1]更新至此寄存器的低7位reg8[6:0],实现右移一位的操作。连续赋值语句把移位寄存器最低位通过qb端口输出。

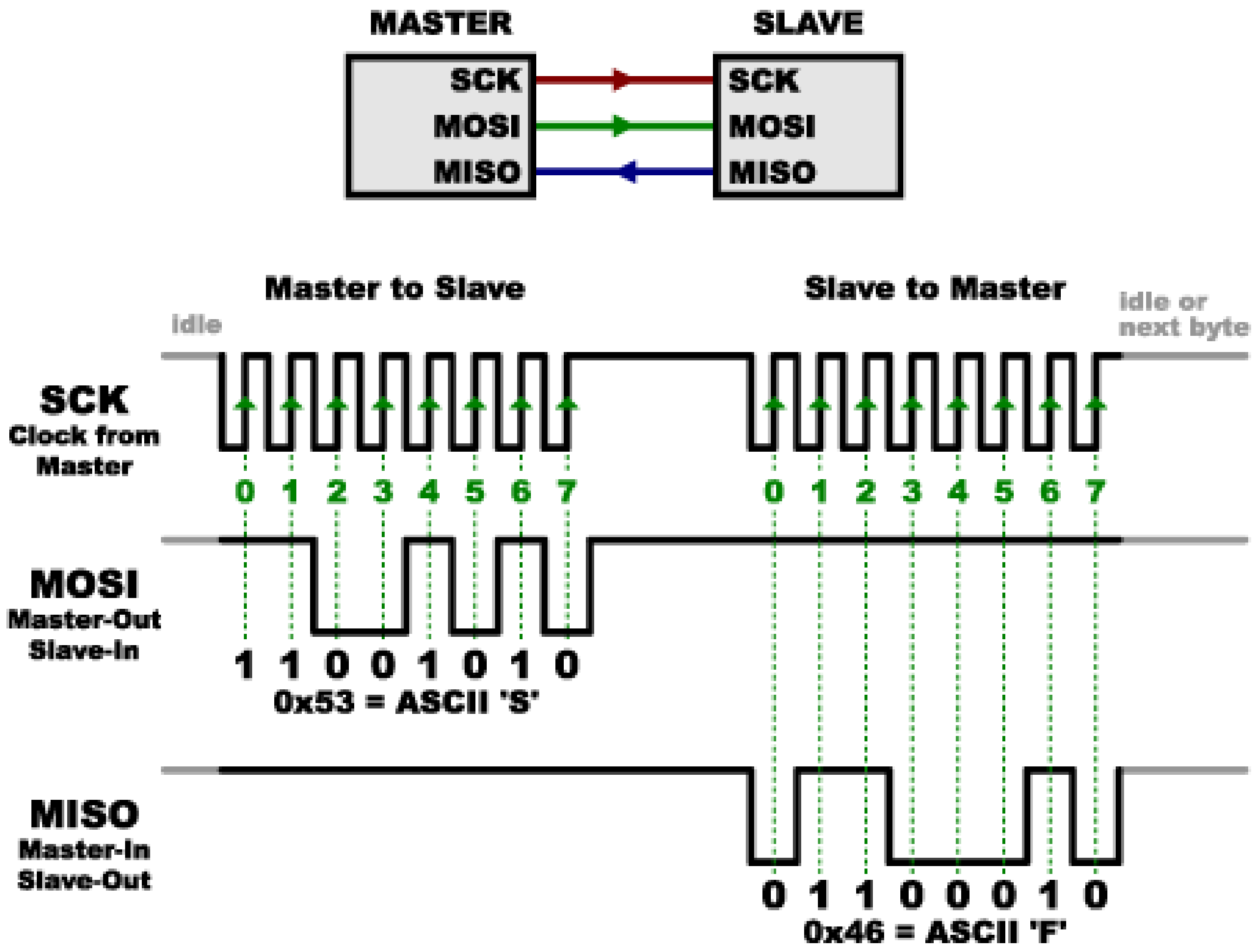


```
`timescale 1ns / 1ps

module shift_reg8
(
    input clk,
    input rst,
    input en,
    input [7:0] din,
    output qb
);

    reg [7:0] reg8;

    always @(posedge clk or negedge rst or negedge en) begin
        if (!rst)
            reg8 <= 0;
        else if(!en)
            reg8 <= din;
        else
            reg8[6:0] <= reg8[7:1];
    end

    assign qb = reg8[0];

endmodule
```

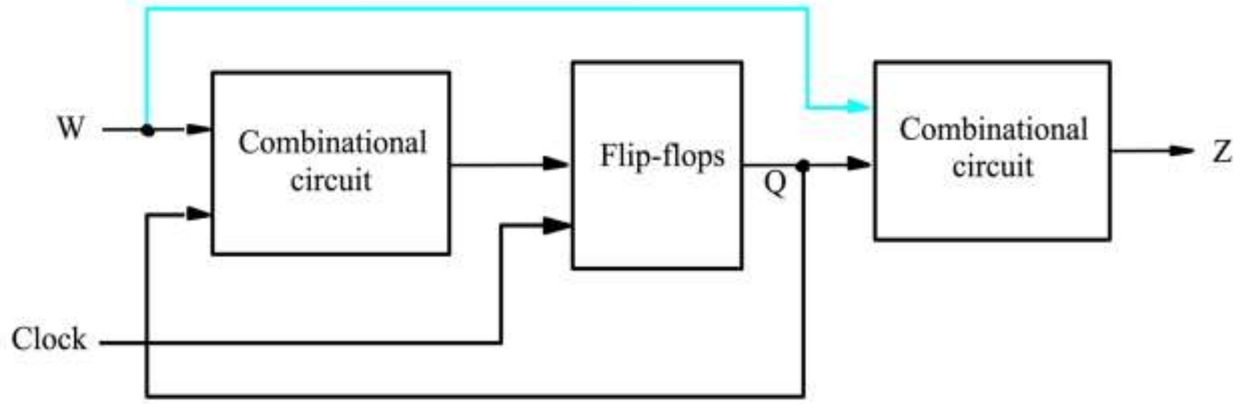
Serial-to-Parallel Interface (SPI)

- 常用串行总线协议（I2C）



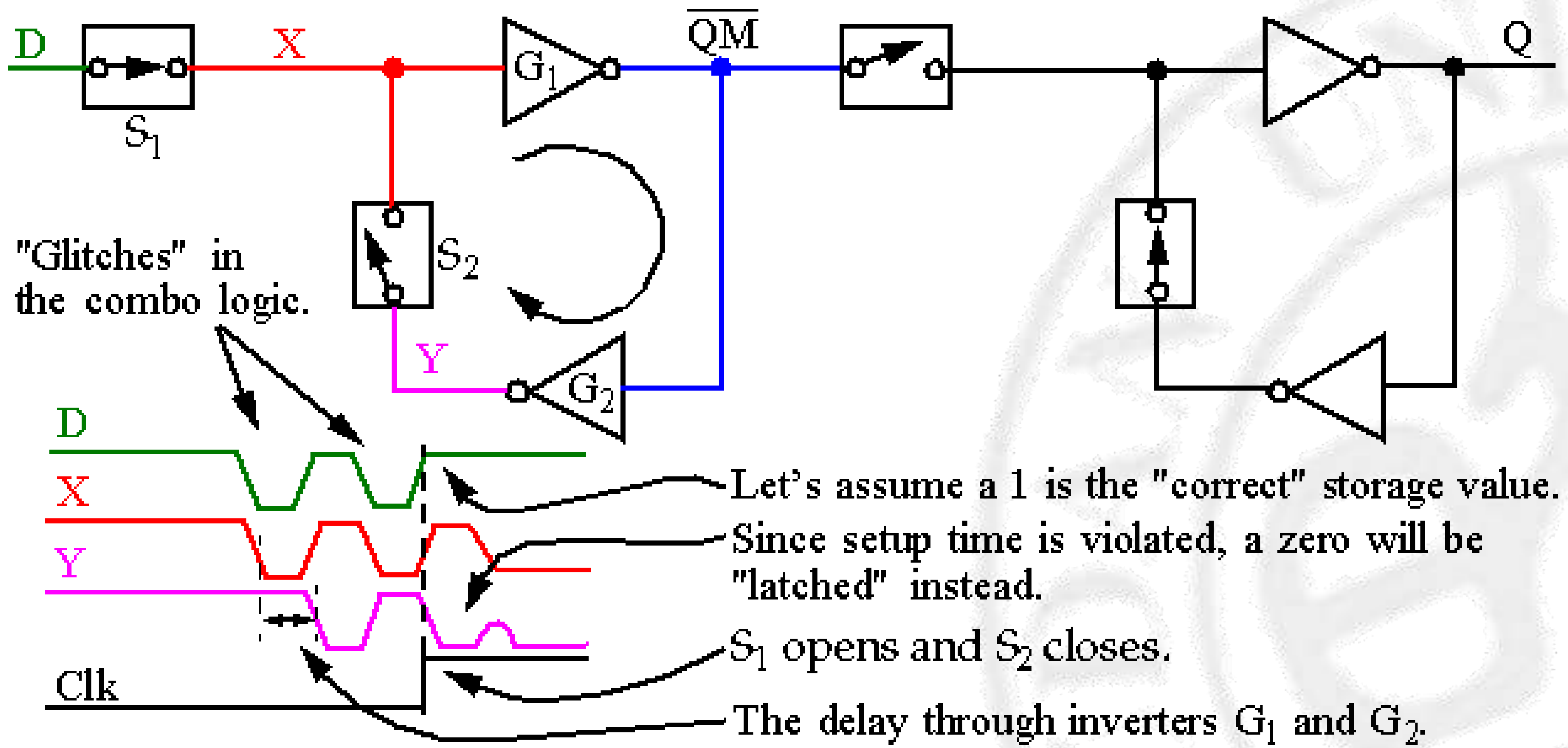


DFF应用(二)同步时序电路

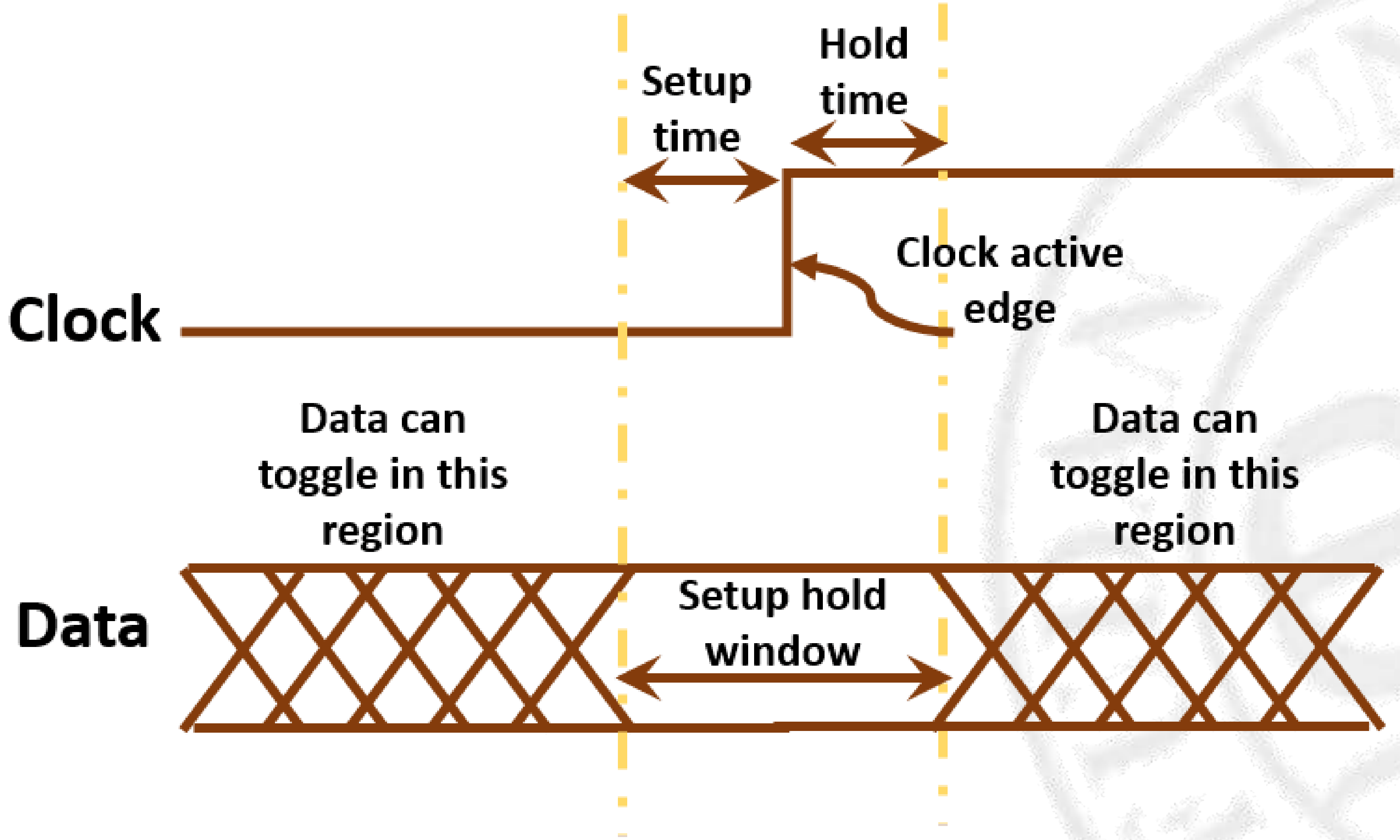


- 数字电路一般分为组合逻辑电路与时序逻辑电路
- 时序逻辑又被分为同步时序电路与异步时序电路，其区别在于前者有统一的时钟源而后者没有
- 同步时序电路相对简单、稳定，大部分设计均采用同步时序电路。

D 触发器的时序要求

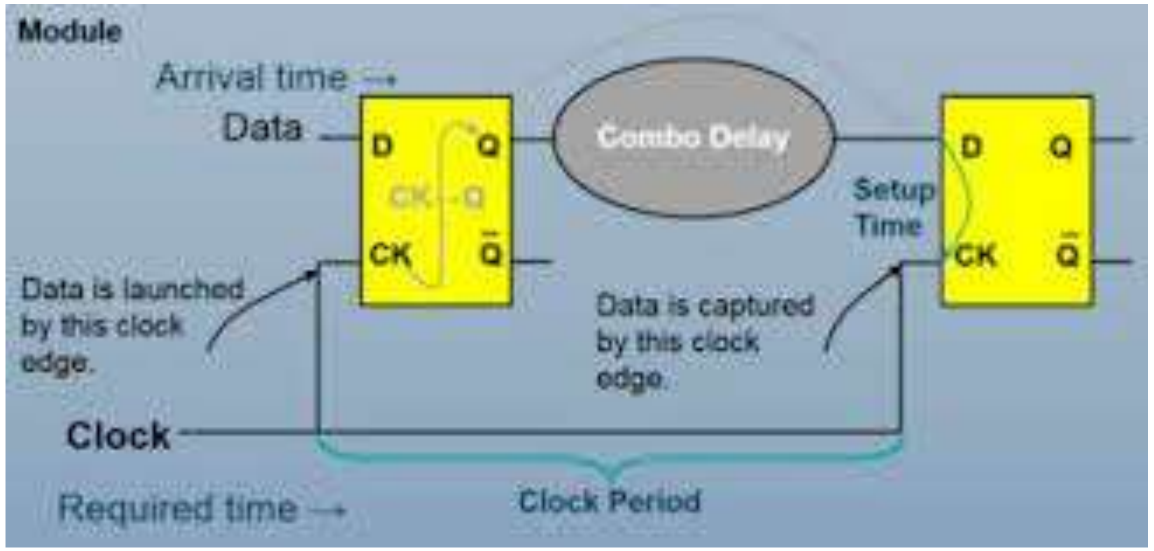

D
X
$\overline{QM}$
Q
S_1
S_2
G_1
G_2
Y
"Glitches" in the combo logic.
D
X
Y
Clk
Let's assume a 1 is the "correct" storage value.
Since setup time is violated, a zero will be "latched" instead.
S_1 opens and S_2 closes.
The delay through inverters G_1 and G_2.

同步时序电路的建立与保持时间

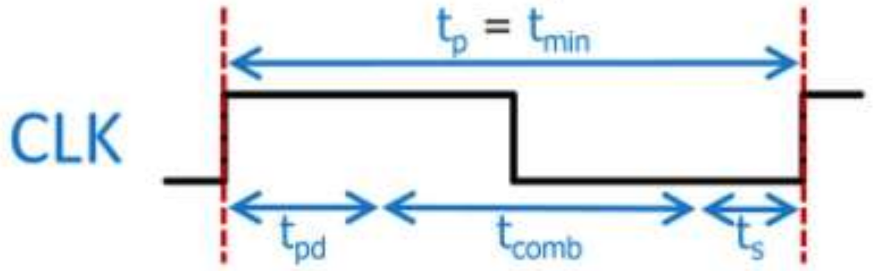


同步时序电路的最小周期

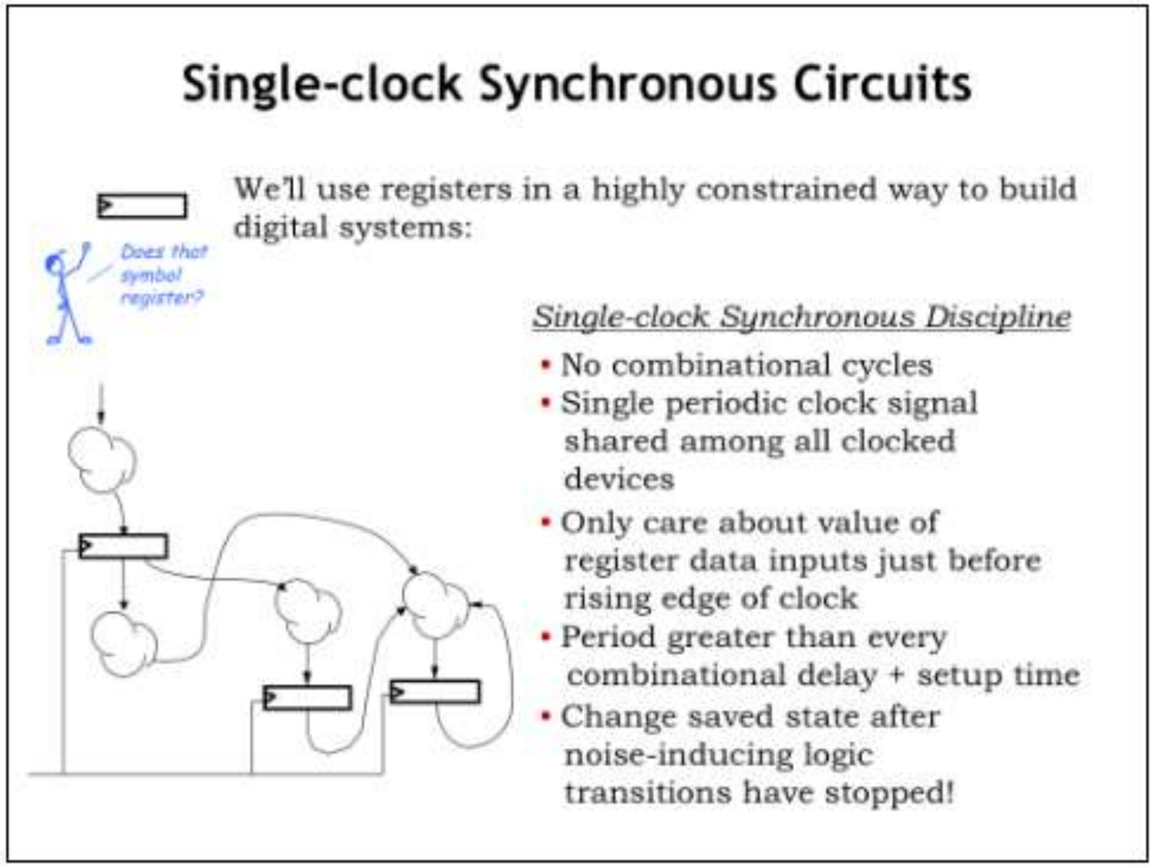
- 原则：下一个上升沿到来前，下一个触发器的D端数据ready



- The minimum clock period is the smallest feasible clock period $t_{min} = t_{pd} + t_{comb} + t_s$
  - Need time for a change to get out of a flip-flop, through the circuit, and be present at another flip-flop input before the start of the set-up time



DFF应用(一)同步时序电路



Single-clock Synchronous Circuits

We'll use registers in a highly constrained way to build digital systems:

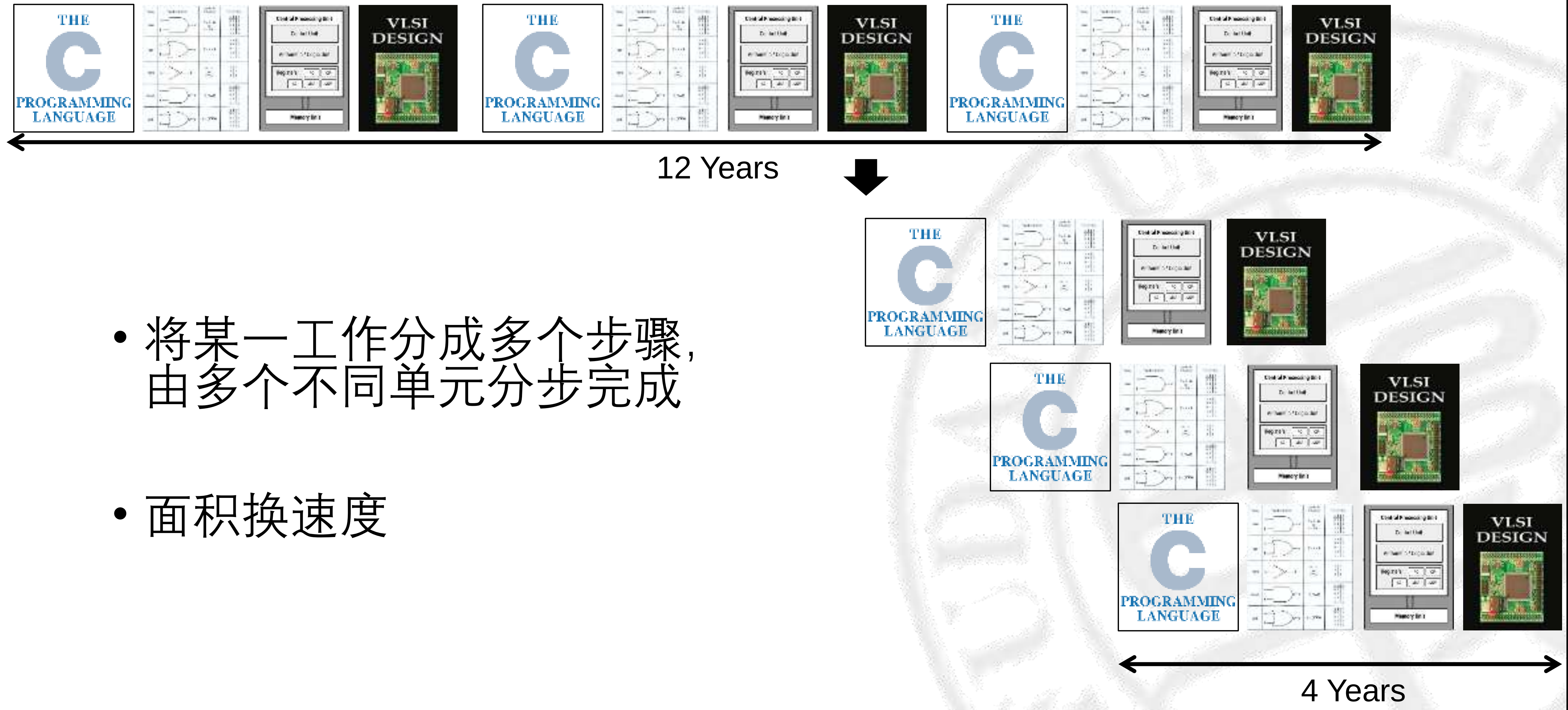
Does that symbol register?

Single-clock Synchronous Discipline

- No combinational cycles
- Single periodic clock signal shared among all clocked devices
- Only care about value of register data inputs just before rising edge of clock
- Period greater than every combinational delay + setup time
- Change saved state after noise-inducing logic transitions have stopped!

- 所有触发器共享同一时钟
- 只关心时钟上升沿之前寄存器输入值
- 时钟周期大于每个组合逻辑延迟加建立时间(大于关键路径)
- 若时钟周期小于关键路径时间，将导致寄存器锁存不稳定错误数据
- 可通过引入流水线寄存器缩短关键路径

DFF应用(二)流水线设计

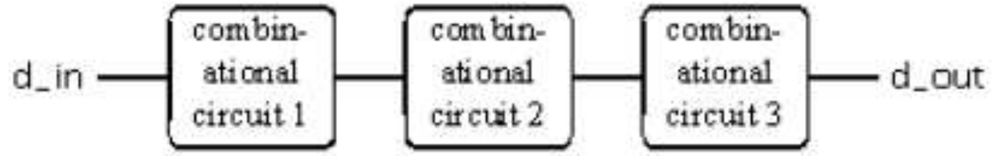


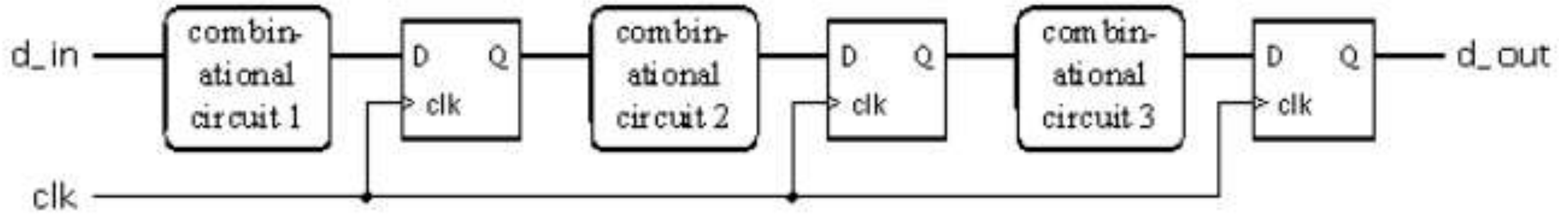
- 将某一工作分成多个步骤，由多个不同单元分步完成
- 面积换速度

基于寄存器（DFF）的流水线设计

Total delay = Delay_1 + Delay_2 + Delay_3

Interval between outputs > Total delay



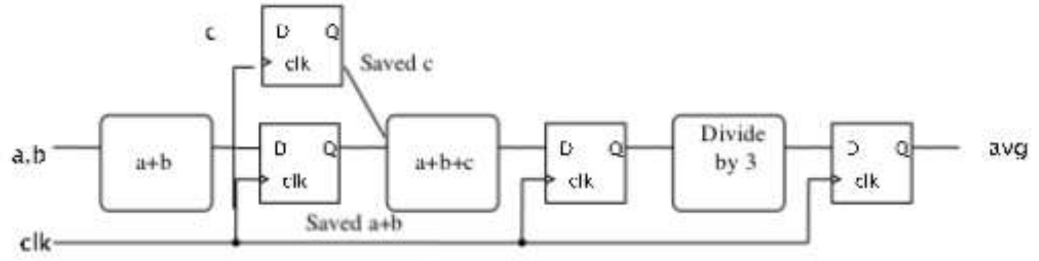


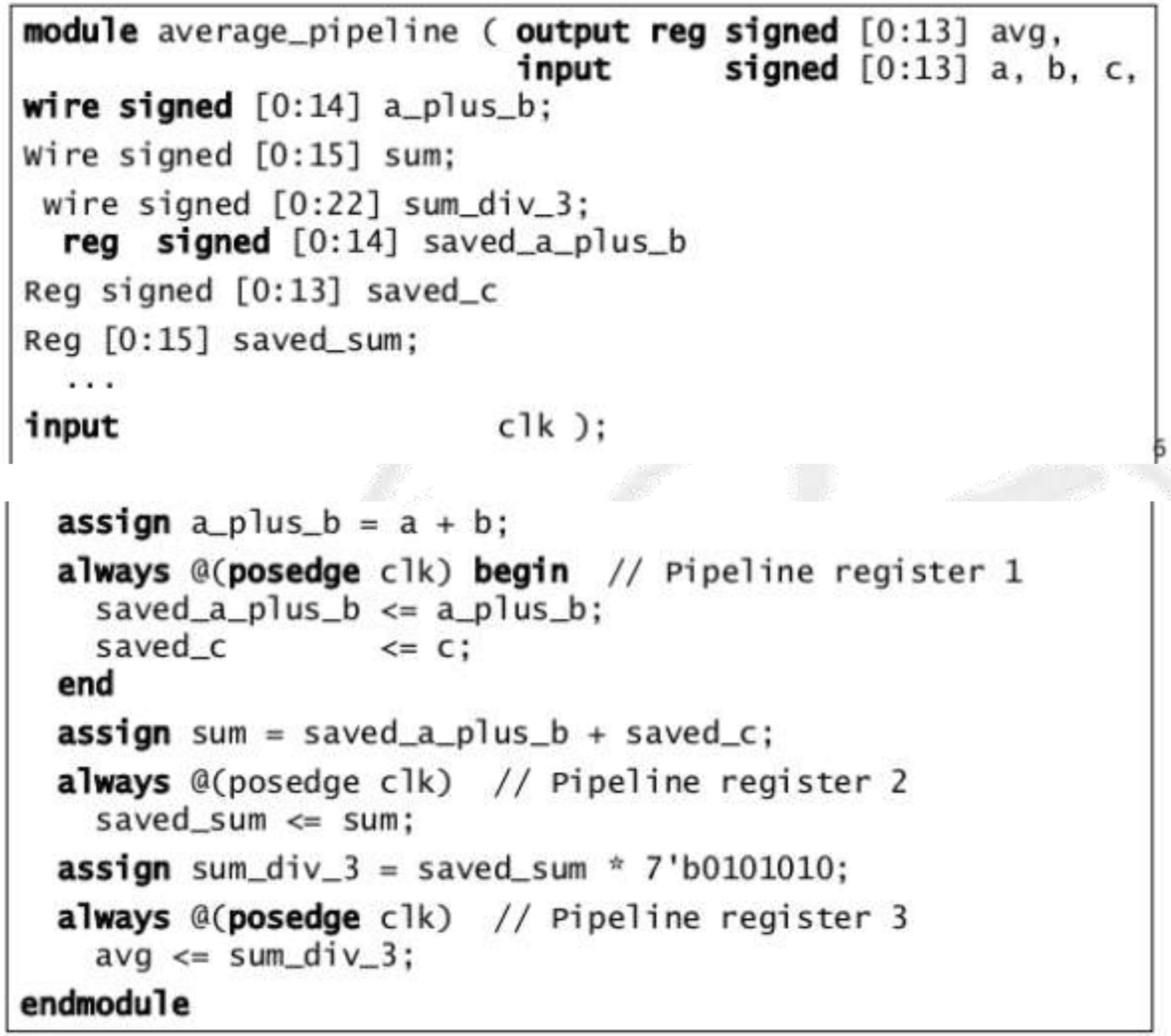
Clock period = max(Delay_1, Delay_2, Delay_3)

Total delay = 3 × clock period

Interval between outputs = 1 clock period

流水线Verilog实例



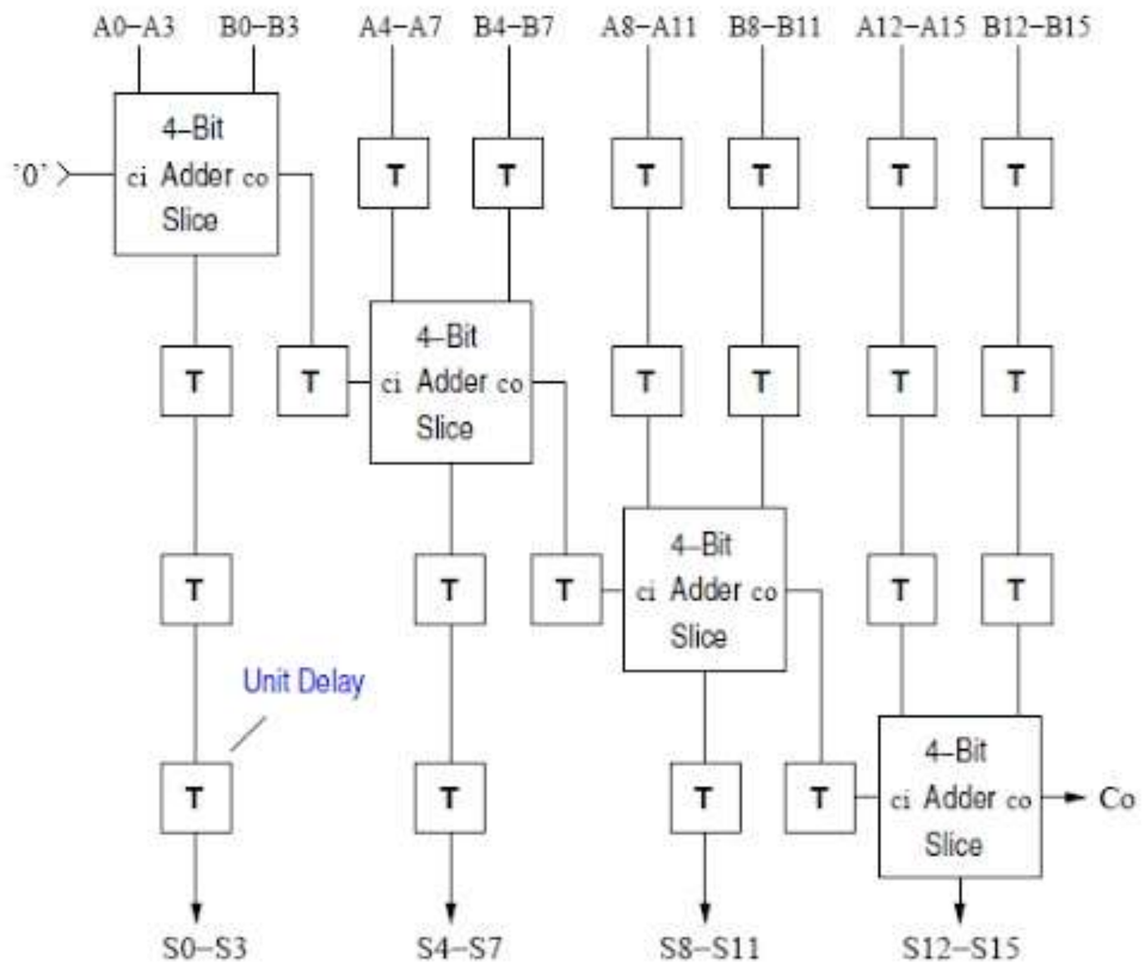


```
module average_pipeline ( output reg signed [0:13] avg,
                          input      signed [0:13] a, b, c,
wire signed [0:14] a_plus_b;
Wire signed [0:15] sum;
 wire signed [0:22] sum_div_3;
  reg  signed [0:14] saved_a_plus_b
Reg signed [0:13] saved_c
Reg [0:15] saved_sum;
   ...
input                         clk );

  assign a_plus_b = a + b;
  always @(posedge clk) begin  // Pipeline register 1
    saved_a_plus_b <= a_plus_b;
    saved_c        <= c;
  end
  assign sum = saved_a_plus_b + saved_c;
  always @(posedge clk)  // Pipeline register 2
    saved_sum <= sum;
  assign sum_div_3 = saved_sum * 7'b0101010;
  always @(posedge clk)  // Pipeline register 3
    avg <= sum_div_3;
endmodule
```

流水线型乘法加法器

- 加法器的级联与进位

```
module power3(
  output reg [7:0] XPower,
  input            clk,
  input      [7:0] X
  );
  reg        [7:0] XPower1, XPower2;
  reg        [7:0] X1, X2;
  always @(posedge clk) begin
    // Pipeline stage 1
    X1      <= X;
    XPower1 <= X;
    // Pipeline stage 2
    X2      <= X1;
    XPower2 <= XPower1 * X1;
    // Pipeline stage 3
    XPower <= XPower2 * X2;
  end
endmodule
```


A0–A3
B0–B3
A4–A7
B4–B7
A8–A11
B8–B11
A12–A15
B12–B15
'0'
4–Bit ci Adder co Slice
T
Unit Delay
Co
S0–S3
S4–S7
S8–S11
S12–S15