

# Laboratory Exercise 3

## Latches, Flip-flops, and Registers

本练习的目的是研究锁存器，触发器和寄存器。

### Part I

Altera FPGA包括可用于实现用户电路的触发器。我们将在本练习的第IV至VII部分中展示如何使用这些流行语。但首先，我们将展示如何在FPGA中创建存储元件，而无需使用其专用的触发器。

图1描绘了门控RS锁存电路。图2给出了使用逻辑表达式描述该电路的VHDL代码样式。如果此锁存器在具有4输入查找表（LUT）的FPGA中实现，则只需要一个查找表，如图3A。

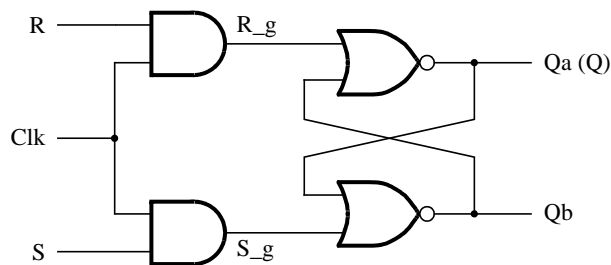


Figure 1. A gated RS latch circuit.

```
-- A gated RS latch described the hard way
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY part1 IS
    PORT ( Clk, R, S : IN    STD_LOGIC;
          Q          : OUT   STD_LOGIC);
END part1;

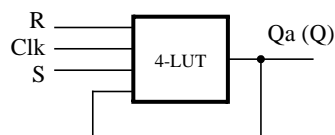
ARCHITECTURE Structural OF part1 IS
    SIGNAL R_g, S_g, Qa, Qb : STD_LOGIC ;
    ATTRIBUTE keep : boolean;
    ATTRIBUTE keep of R_g, S_g, Qa, Qb : SIGNAL IS true;
BEGIN
    R_g <= R AND Clk;
    S_g <= S AND Clk;
    Qa <= NOT (R_g OR Qb);
    Qb <= NOT (S_g OR Qa);

    Q <= Qa;

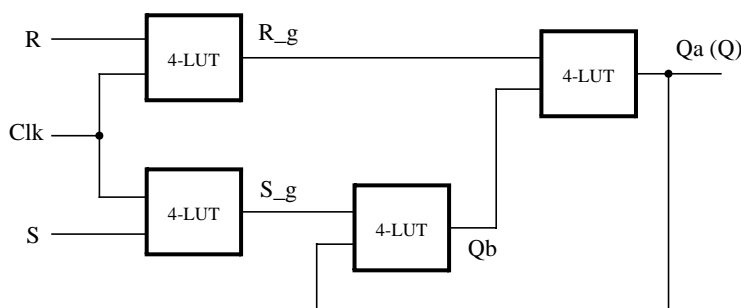
END Structural;
```

Figure 2. Specifying the RS latch by using logic expressions.

虽然可以在一个4输入LUT中正确地实现锁存器，但是这种实现方式不允许观察其内部信号，例如R\_g和S\_g，因为它们不是作为LUT的输出提供的。为了在实现的电路中保留这些内部信号，必须在代码中包含编译器指令。在图2中，使用VHDL ATTRIBUTE语句包含指令keep；它指示Quartus II编译器为每个信号R\_g, S\_g, Qa和Qb使用单独的逻辑元素。编译代码会产生具有四个4-LUT的电路，如图3b所示。



(a) Using one 4-input lookup table for the RS latch.



(b) Using four 4-input lookup tables for the RS latch.

Figure 3. Implementation of the RS latch from Figure 1.

为RS锁存电路创建Quartus II项目，如下所示：

1. 为RS锁存器创建一个新项目。选择Cyclone II EP2C35F672C6作为目标芯片，这是Altera DE2板上的FPGA芯片。
2. 使用图2中的代码生成VHDL文件并将其包含在项目中。
3. 编译代码。使用Quartus II RTL Viewer工具检查代码生成的门级电路，并使用Technology Viewer工具验证锁存器是否如图3b所示实现。
4. 创建一个矢量波形文件（.vwf），它指定电路的输入和输出。绘制R和S输入的波形，并使用Quartus II Simulator为R\_g, S\_g, Qa和Qb生成相应的波形。使用功能和时序仿真验证锁存器是否按预期工作。

## Part II

图4显示了门控D锁存器的电路。

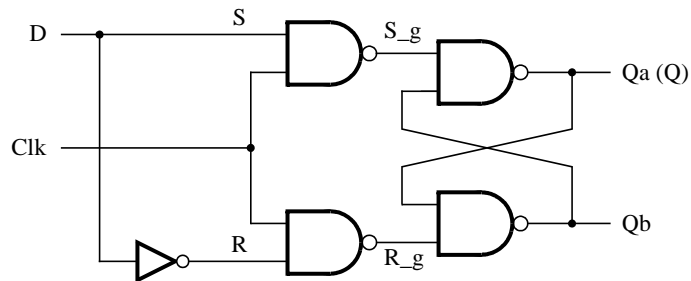


Figure 4. Circuit for a gated D latch.

执行以下步骤：

1. 创建一个新的Quartus II项目。使用图2中的代码样式为门控D锁存器生成VHDL文件。使用keep指令确保使用单独的逻辑元素来实现信号R, S\_g, R\_g, Qa和Qb。
2. 选择Cyclone II EP2C35F672C6作为目标芯片并编译代码。使用Technology Viewer工具检查已实现的电路。
3. 通过使用功能仿真验证锁存器是否适用于所有输入条件。使用时序仿真检查电路的时序特性。
4. 创建一个新的Quartus II项目，该项目将用于实现DE2板上的门控D锁存器。该项目应包含一个顶层实体，其中包含DE2板的相应输入和输出端口（引脚）。在此顶级实体中实例化您的锁存器。使用开关SW<sub>0</sub>驱动锁存器的D输入，并使用SW<sub>1</sub>作为Clk输入。将Q输出连接到LEDR<sub>0</sub>。
5. 重新编译项目并将编译好的电路下载到DE2板上。
6. 通过切换D和Clk开关并观察Q输出来测试电路的功能。

### Part III

图5显示了主从D触发器的电路。

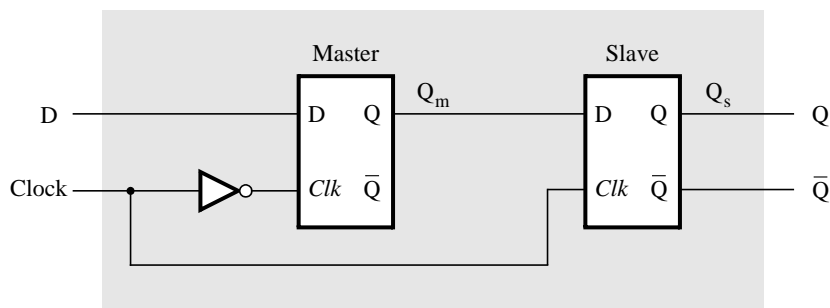


Figure 5. Circuit for a master-slave D flip-flop.

执行以下操作：

1. 创建一个新的Quartus II项目。生成一个VHDL文件，从第二部分实例化您的门控D锁存器实体的两个副本，以实现主从传输。

2. 在项目中包含Altera DE2板的相应输入和输出端口。使用开关SW<sub>0</sub>驱动触发器的D输入，并使用SW<sub>1</sub>作为时钟输入。将Q输出连接到LEDR<sub>0</sub>。
3. 编译您的项目。
4. 使用Technology Viewer检查D flip-flop电路，并使用仿真来验证其正确的操作。
5. 将电路下载到DE2板上，通过切换D和时钟开关并观察Q输出来测试其功能。

#### Part IV

图6显示了一个具有三种不同存储元件的电路：门控D锁存器，正边沿触发D触发器和负边沿触发D触发器。

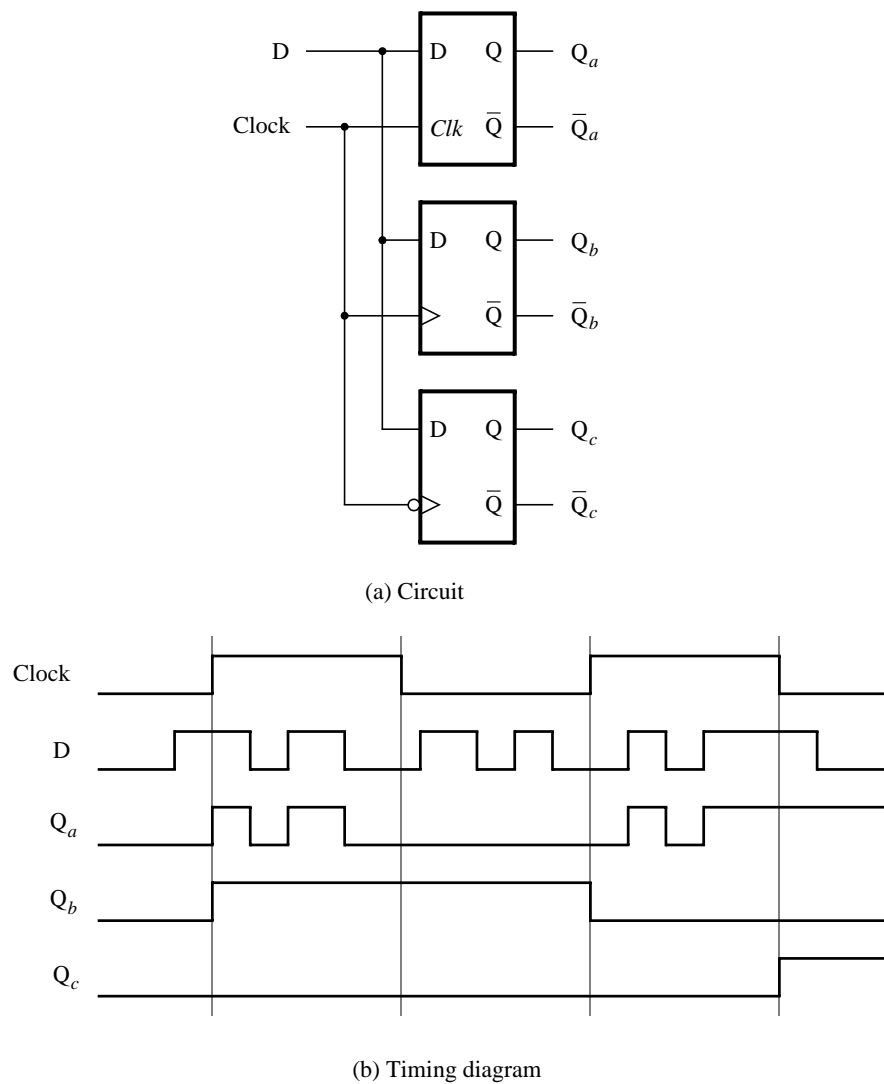


Figure 6. Circuit and waveforms for Part IV.

使用Quartus II软件实现和模拟该电路如下：

1. 创建一个新项目。
2. 编写一个实例化三个存储元素的VHDL文件。对于此部分，您不应再使用第I部分到第III部分中的keep指令（即VHDL ATTRIBUTE 语句）。图7给出了VHDL代码的行为风格，该代码指定了图4中的门控D锁存器。该锁存器可以在一个4输入查找表中实现。使用类似的代码样式来指定图6中的flip-flop操作。
3. 编译代码并使用Technology Viewer检查实现的电路。验证锁存器是否使用一个查找表，并使用目标FPGA中提供的触发器来实现触发器操作。
4. 创建一个矢量波形文件（.vwf），它指定电路的输入和输出。绘制输入D和Clock，如图6所示。使用功能仿真获得三个输出信号。观察三个存储元件的不同行为。

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY latch IS

    PORT ( D, Clk : IN    STD_LOGIC ;
          Q       : OUT   STD_LOGIC );
END latch ;

ARCHITECTURE Behavior OF latch IS
BEGIN
    PROCESS ( D, Clk )
    BEGIN
        IF Clk = '1' THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

Figure 7. A behavioral style of VHDL code that specifies a gated D latch.

## Part V

我们希望在四个7段显示器HEX<sub>7.4</sub>上显示16位数字A的十六进制值。我们还希望在四个7段显示器上显示16位数字B的十六进制值，HEX<sub>3.0</sub>。A和B的值是电路的输入，它们通过开关SW<sub>15.0</sub>提供。首先将开关设置为A的值，然后将开关设置为B的值，即可完成此操作。因此，A的值必须存储在电路中。

1. 创建一个新的Quartus II项目，用于在Altera DE2板上实现所需的电路。
2. 编写提供必要功能的VHDL文件。使用KEY<sub>0</sub>作为低电平有效的异步复位，并使用KEY<sub>1</sub>作为时钟输入。在项目中包含VHDL文件并编译电路。
3. 分配FPGA上的引脚以连接到开关和7段显示器，如DE2板的用户手册中所示。
4. 重新编译电路并将其下载到FPGA芯片中。
5. 通过切换开关并观察输出显示来测试设计的功能。