流水灯实验

201609160125 16级通信工程一班 袁春清

**要求**：

1. 上电后4个灯全亮。
2. 1S之后4个灯全灭。
3. 1S之后顺序点亮LED1，led2，led3，led4。
4. 1s之后顺序熄灭led1，led2，led3，led4。
5. 1s之后4个灯全亮。

实验说明：

本实验是在黑金FPGA开发板的资料01\_run\_led添加代码而来的，例程的位置（VERILOG\1\_module\Experiment\01\_run\_led），只在源例程上添加代码，无其他改动。

思路分析：

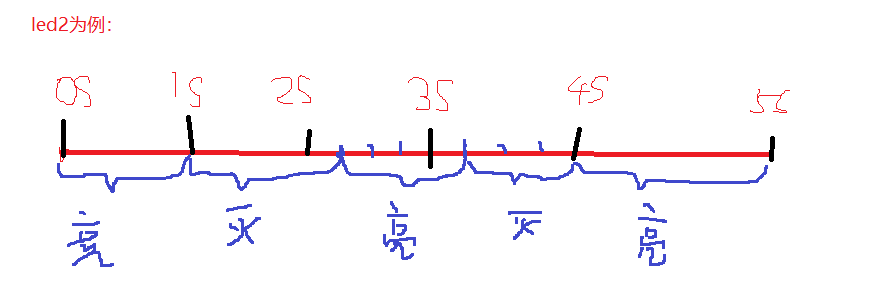
FPGA板载晶振为f = 50MHz,所以一个时钟周期T =1/f = 1/(50M) 单位S，即为一个时钟周期时间是 20ns, 如果要计时1S，需要50\_000\_000个时钟周期。实验要求如上所示，思路是实验分为四个module，每个灯对应一个module，在module中实现控制Led灯状态。最后实例化。

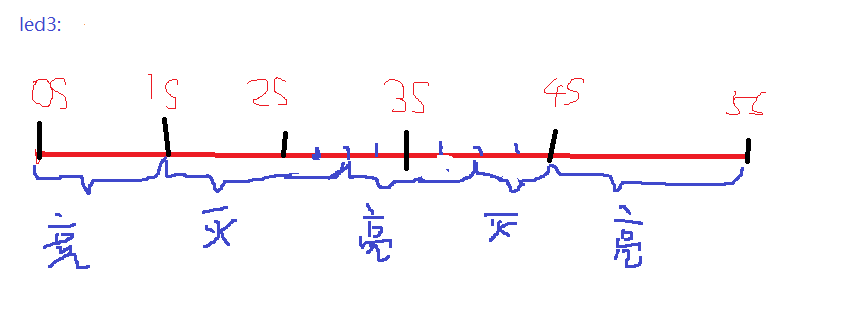
Count1 为计数变量，随时钟加一，

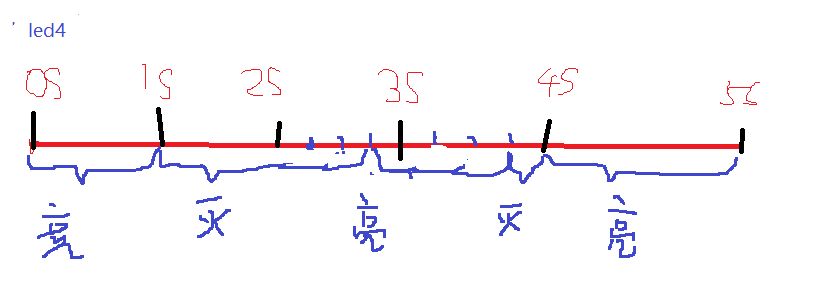
思路见代码注释和时间流程图了。

**Led1~4 时间流程图：**

led1图简单不画了。







代码示例：

Module0:

always @ ( posedge CLK or negedge RSTn )

if( !RSTn )

rLED\_Out <= 1'b0;

else if( Count1 >= 28'd0 && Count1 < 28'd50\_000\_000 )//在第0到一秒，led0亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd50\_000\_000 && Count1 < 28'd100\_000\_000 ) //在第一到二秒，led0灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd100\_000\_000 && Count1 < 28'd150\_000\_000 )//在第二秒到点三秒led0亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd150\_000\_000 && Count1 < 28'd200\_000\_000 )//在第三秒到第四秒led0灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd200\_000\_000 && Count1 < 28'd250\_000\_000 )//同的第一步

rLED\_Out <= 1'b1;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

assign LED\_Out = rLED\_Out;

Module1:

always @ ( posedge CLK or negedge RSTn )

if( !RSTn )

rLED\_Out <= 1'b0;

else if( Count1 >= 28'd0 && Count1 < 28'd50\_000\_000 )//在第0到一秒，led1亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd50\_000\_000 && Count1 < 28'd112\_500\_000 ) //在第一到二又四分之一秒灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd112\_500\_000 && Count1 < 28'd162\_500\_000 )//在二又四分之一到三又四分之一秒亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd162\_500\_000 && Count1 < 28'd200\_000\_000 )//在三又四分之一到四秒灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd200\_000\_000 && Count1 < 28'd250\_000\_000 )//四到五秒亮

rLED\_Out <= 1'b1;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

assign LED\_Out = rLED\_Out;

Module2:

always @ ( posedge CLK or negedge RSTn )

if( !RSTn )

rLED\_Out <= 1'b0;

else if( Count1 >= 28'd0 && Count1 < 28'd50\_000\_000 )//在第一秒，亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd50\_000\_000 && Count1 < 28'd125\_000\_000 ) //在第一到二又四分之二秒灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd125\_000\_000 && Count1 < 28'd175\_000\_000 )//在二又四分之二到三又四分之二亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd175\_000\_000 && Count1 < 28'd200\_000\_000 )//在三又四分之二到四秒灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd200\_000\_000 && Count1 < 28'd250\_000\_000 )//四秒到五秒亮

rLED\_Out <= 1'b1;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

assign LED\_Out = rLED\_Out;

Module3:

always @ ( posedge CLK or negedge RSTn )

if( !RSTn )

rLED\_Out <= 1'b0;

else if( Count1 >= 28'd0 && Count1 < 28'd50\_000\_000 )//在第一秒，亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd50\_000\_000 && Count1 < 28'd137\_500\_000 ) //在第一到二又四分之三秒，灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd137\_500\_000 && Count1 < 28'd187\_500\_000 )//在二又四分之三到三又四分之三秒亮

rLED\_Out <= 1'b1;

else if( Count1 >=28'd187\_500\_000 && Count1 < 28'd200\_000\_000 )//三又四分之三到四秒灭

rLED\_Out <= 1'b0;

else if( Count1 >=28'd200\_000\_000 && Count1 < 28'd250\_000\_000 )//四到五秒灭

rLED\_Out <= 1'b1;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

assign LED\_Out = rLED\_Out;

///////////////////////////////////////////////////////////////////////////////////////////////

// By----程舒德，袁春清