

竞赛题目：基于FPGA的三子棋双人博弈

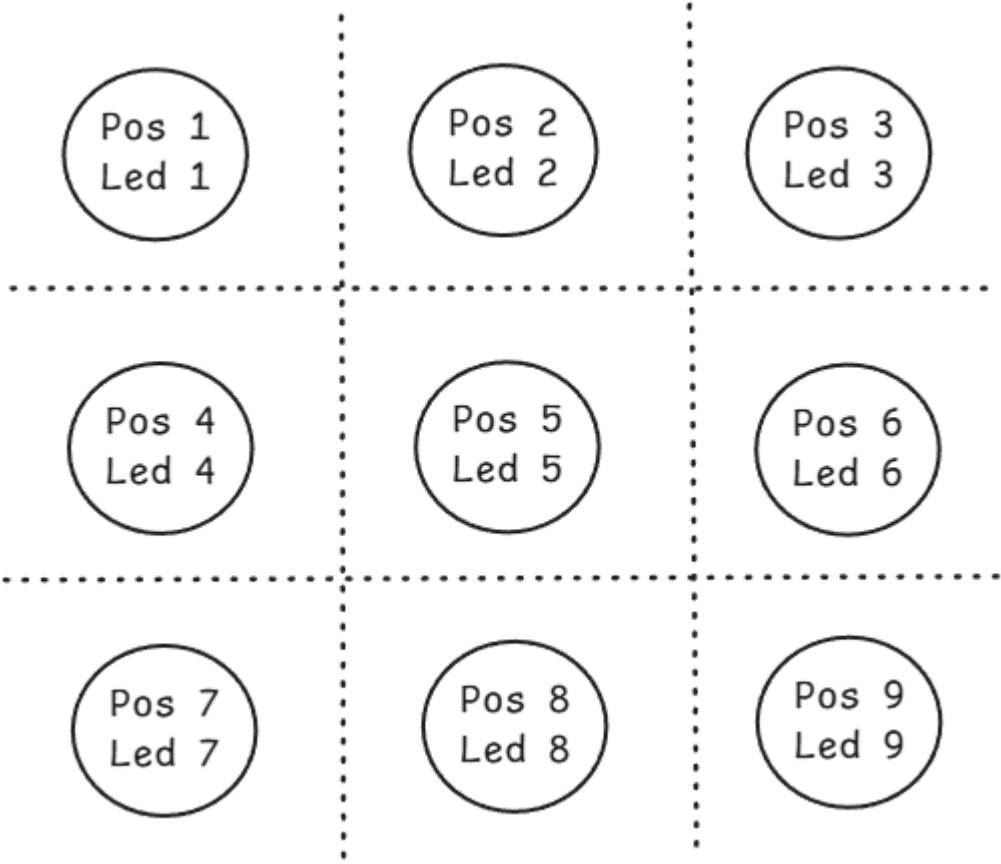
背景介绍：

三子棋（Tic-Tac-Toe）是一款经典的雙人游戏，本次竞赛要求选手设计并实现一个基于Verilog的三子棋游戏，其中玩家将与计算机进行对战。参赛者需要设计电路来控制游戏流程，检测游戏胜负，并通过LED指示落子情况。本次竞赛不仅考验选手的Verilog编程能力，还要求选手具备数字逻辑电路设计的综合能力。

游戏规则：

1. 棋盘：

- 棋盘为3x3的网格，共有9个位置。每个位置用2位二进制数表示状态：
 - 2'b00：该位置为空，未被占用。
 - 2'b01：玩家在该位置落子（X）。
 - 2'b10：计算机在该位置落子（O）。
- 棋盘每个棋子的位置和LED灯的位置如下图所示：



2. 游戏流程：

- 玩家和计算机轮流落子，每回合玩家按下按钮进行落子，计算机自动选择空位落子。
- 玩家落子后，该位置的红色LED亮起；计算机落子后，该位置的绿色LED亮起。
- 当玩家或计算机成功在一行、一列或对角线连续放置三个相同的符号时，该方获胜。

3. 胜利检测：

- 以下行列或对角线连续三个相同符号将判定为胜利：
 - 行：(1,2,3)，(4,5,6)，(7,8,9)
 - 列：(1,4,7)，(2,5,8)，(3,6,9)
 - 对角线：(1,5,9)，(3,5,7)

4. 非法移动检测：

- 若玩家或计算机试图在已被占用的位置落子，则判定为非法移动，并提示重新选择位置。

5. 平局检测：

- 若所有位置均已被占用且没有一方获胜，则判定为平局。

基本任务要求：

1. 棋盘实现： 使用Verilog代码实现3x3棋盘，能够存储每个位置的状态（空位、玩家X或计算机O）。
2. 胜利检测电路： 设计胜利检测逻辑，当满足胜利条件时，电路输出胜利信号。
3. 非法移动检测： 设计非法移动检测逻辑，防止在已被占用的位置重复落子。
4. 平局检测： 实现平局检测逻辑，当所有位置都被占用且无胜利者时，输出平局信号。
5. LED显示： 使用Verilog代码控制棋盘上每个位置的LED。当玩家或计算机落子时，点亮相应的红色或绿色LED。

提交要求：

- 需要修改的内容： Verilog代码文件：tic_tac_toe.v 中所有设计模块及顶层模块，tic_tac_toe.xdc 文件的时钟频率。
- 不可以修改的内容： tb_tic_tac_toe.v, tic_tac_toe_script.tcl
- 自动生成的内容： reports文件夹的文件
- 不可以提交的内容： tic_tac_toe_prj文件夹

- 可以参考的内容：golden文件夹
- 提交格式：

```
tic_tac_toe/  
|-- golden/  
|   |-- sim_results_behav_golden.log  
|-- reports/  
|   |-- sim_results_behav.log  
|   |-- sim_waveform_behav.wcfg  
|   |-- sim_waveform_behav.wdb  
|   |-- sim_waveform_time_synth.wcfg  
|   |-- sim_waveform_time_synth.wdb  
|   |-- timing_report.rpt  
|   |-- utilization_report.rpt  
|-- src/  
|   |-- tic_tac_toe.v  
|-- sim/  
|   |-- tb_tic_tac_toe.v  
|-- xdc/  
|   |-- tic_tac_toe.xdc  
|-- tic_tac_toe_script.tcl
```

评分标准：

- 功能实现（40分）：
 - 输赢检测正确（15分）
 - 平局检测正确（15分）
- 综合时序表现与资源利用率（10分）