

-
- ```

graph TD
 DE1_SoC[DE1-SoC] --> clock_divider[clock_divider]
 DE1_SoC --> LEDDriver[LEDDriver]
 clock_divider --> d_ff[d-ff]
 LEDDriver --> User_Input[User Input]
 d_ff --> color[color]
 User_Input --> color
 User_Input --> Column_Input[Column Input]
 User_Input --> animated_Start[animated Start]
 color --> Column_Input
 Column_Input --> animated_Start
 animated_Start --> animated_LED[animated LED]
 animated_LED --> centerPoint[centerPoint]
 centerPoint --> Victory_module[Victory module]
 Victory_module --> p2HorizontalVictory[p2HorizontalVictory]
 Victory_module --> p1HorizontalVictory[p1HorizontalVictory]
 Victory_module --> p2VerticalVictory[p2VerticalVictory]
 Victory_module --> p1VerticalVictory[p1VerticalVictory]
 Victory_module --> p2DiagonalVictory[p2DiagonalVictory]
 Victory_module --> p1DiagonalVictory[p1DiagonalVictory]
 Victory_module --> p2Diagonal2Victory[p2Diagonal2Victory]
 Victory_module --> p1Diagonal2Victory[p1Diagonal2Victory]

```
- DE1-SoC ← Top-level module
- clock\_divider, LEDDriver
- User Input ← SW[9] reset switch  
KEY[3] and KEY[2] moves column placement (left and right)  
KEY[1] drops piece.
- d-ff
- color ← When KEY[1] is hit, swap to other player's turn/swap turn.
- Column Input
- animated Start ← Top row shows LED of column piece placement
- animated LED ← Drops the animated piece
- centerPoint
- Victory module ← Victory module with 8 win conditions (4 per player)
- p2HorizontalVictory, p1HorizontalVictory, p2VerticalVictory, p1VerticalVictory, p2DiagonalVictory, p1DiagonalVictory, p2Diagonal2Victory, p1Diagonal2Victory
- Combinational logic issued for all victory conditions in DE1-SoC module.