**1. Position Decoder:**

When user finally selects the position, the Position Decoder Module converts user selected position and returns the output position on as per grid.

**1.1 Inputs:**

**Enable Bit** – Enable Bit will work as active high and it enables the decoder when Enable Bit is 1, otherwise, disables it.

# of Bits in Enable Bit = 1

**Position Input** – Position Input will contain user selected position in 4 bits. For example, if user selects tile no ‘10’ on grid, then the position input will be ‘1010’.

# of Bits in Position Input = 4

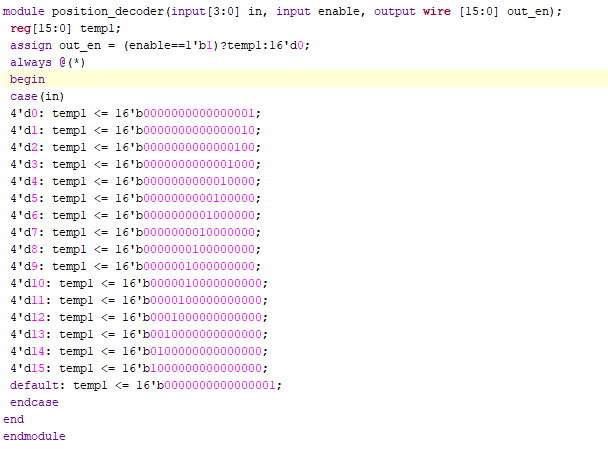
**1.2 Outputs:**

**Decoded Position** – It is 16 bits binary number. All bits of Decoded Position will be ‘0’ except the nth bit where nth bit will be decided as per position input decimal value. For example, if position input is ‘1010’, then it’s ‘10’ in decimal. Hence, the 10th LSB bit will be ‘1’.

e.g ‘000001000000000’

# of Bits in Decoded Position = 16

**1.3 Code Snippets:**



**2. Win Detector:**

Win Detector Module check for every possible winning combination of current grid after every valid change in position. If any player wins the match, he will be assigned as winner.

**1.1 Inputs:**

**All Position Inputs** – It will contain 16 inputs. These 16 inputs are 2 bits position inputs and contains three states: “X” , “O” and the other one is undefined (i.e when the tile is not filled because it is possible that tiles may not be filled and any player won the game).

# of Bits in Position Inputs Bit = 2

**1.2 Outputs:**

**Winner** – Return ‘1’ if anyone is winner else returns ‘0’

# of Bits in Winner = 1

**WinnerID** – Since there are two players: Player 1 and Player 2. The Player 1 ID is ‘01’ and Player 2 ID is ‘10’. It returns any of the Player ID who wins the game.

# of Bits in WinnerID = 2