SJSU EE287 Spring 2020 project

This semester, you will be designing the permutation engine for the sha3-256 hash. The engine has 1600 bits in, performs 24 levels of permutation with each level having multiple steps. The logic performed involved bit moving (Which takes little to no gates) , XORs, and AND operations.
The design will have a 4.65ns cycle time. The interface has 200 bits in per push, and takes 8 clocks to load the 1600 bits. There is a 3 bit code describing which 200 bit portion is on the interface. The entire 1600 bits have been loaded with dix==7. No other ordering is guaranteed. The output interface is 200 bits, and requires 8 clocks to send the output. The output index must go from 0 to 7 on output. This leaves 8 cycles for the calculation. (3 levels per clock).

The test bench requires the code be placed in file ‘perm.sv’ .

The data transformation is found in the attached specification. The data is in a 3D array. I used a set of packed dimensions [4:0][4:0][63:0]. The first dimension is the row, the second the column, and the third is the bit.

The interface:

```
module perm(input clk, input reset, input [2:0] dix,input [199:0] din,
            input pushin, output [2:0] doutix, output [199:0] dout,
            output pushout);
```