HW1 – EE272 F20

You are designing a box to do the sha3 permutation as described in section 3.2 of the NIST FIPS PUB 202. The design shall have w=64, and the data is 5x5x64. The design has limitations. You **MUST** use 4 provided memory blocks (provided, you instantiate them in the design). The design has a 64 bit input interface and a 64 bit output interface. The design is **limited to 400 flip-flops** not including the external memories. The design is not high performance. It should be able to receive inputs, perform calculations, and send output all at the same time. (With some limits of 25 clocks or so to move data from one memory to another) It will require a rather complex state machine as each permutation requires 24 rounds of 6 steps each with each step reading and writing 25 memory locations.

The module **MUST** have the following module header:

```
module perm_blk(input clk, input rst, input pushin, output reg stopin,
        input firstin, input [63:0] din,
        output reg [2:0] m1rx, output reg [2:0] m1ry,
        input [63:0] m1rd,
        output reg [2:0] m1wx, output reg [2:0] m1wy,output reg m1wr,
        output reg [63:0] m1wd,
        output reg [2:0] m2rx, output reg [2:0] m2ry,
        input [63:0] m2rd,
        output reg [2:0] m2wx, output reg [2:0] m2wy,output reg m2wr,
        output reg [63:0] m2wd,
        output reg [2:0] m3rx, output reg [2:0] m3ry,
        input [63:0] m3rd,
        output reg [2:0] m3wx, output reg [2:0] m3wy,output reg m3wr,
        output reg [63:0] m3wd,
        output reg [2:0] m4rx, output reg [2:0] m4ry,
        input [63:0] m4rd,
        output reg [2:0] m4wx, output reg [2:0] m4wy,output reg m4wr,
        output reg [63:0] m4wd,
        output reg pushout, input stopout, output reg firstout, output reg [63:0] dout);
```

The design is clocked on the positive edge of clk, and the reset is active high. The memory block is called m55.sv It has two ports. One for read, and one for write. Each port has two 3 bit addresses. Each address ranges from 0-4. addresses 5-7 on each address result in X on the memory output.

The design **MUST** be in a file called **perm.sv**

**The design MUST be synthesized!!!**

A test bench tbpm.sv is provided. If you create any additional module files, include them with a `include statement in perm.sv

You can run your code using ./sv_vcs tbpm.sv

A script will be provided for synthesis…

Hints and suggestions:

The four memories can be used as follows

- Input buffer (1)
- Working memories (2)
- Output buffer (1)

It is possible to combine Rho and Pi steps, but you have to do some mapping.

A large text file is used by the test bench.  The file has expected values for each of the 24 rounds for each step.  This can help with debug.