Assignment - synchronous FIFO

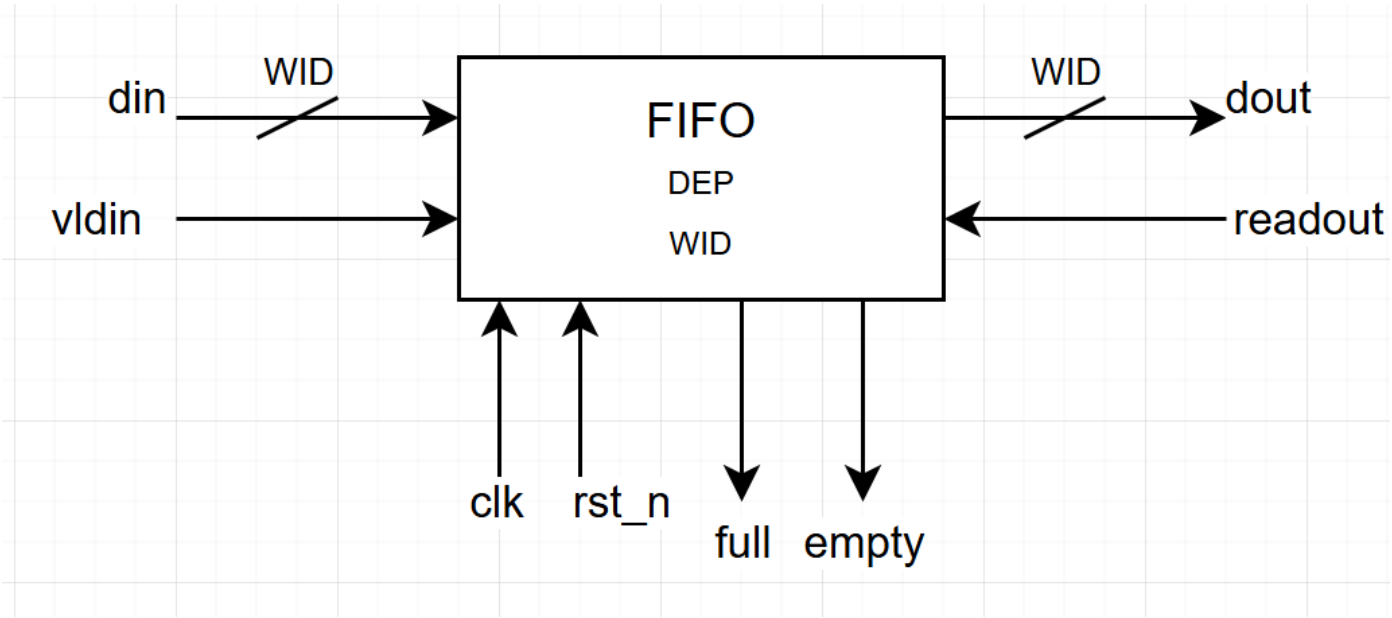
In this assignment you will be implementing a FIFO queue.

Inputs:
clk - clock
rst_n - reset our queue.
parameter - **WID** - width: size of elements to save in queue.
parameter - **DEP** - depth: queue size.
din[WID-1:0] - the data we want to push in our queue.
vldin - when turned on we should save **din[WID-1:0]** in our queue.
readout - pop the FIFO queue and remove it from the queue.
Note: readout when queue is empty should return all zeros.

outputs:
empty - queue is empty.
full- we have reached **DEP** elements and queue is full.
Note: if full is on, we should ignore vldin and we cannot push any more elements to queue.
dout - outputs the "oldest" element in FIFO queue.

parameters

- **WID** : define width of the data bus.
- **DEP** : define the depth of the fifo, that is the number of physical entries in the fifo.



Here is a headstart:

```
module FIFO #(parameter WID = 8, DEP = 16)(
    input clk
    ,input rst_n
    ,input din[WID-1:0]
    ,input vldin
    ,input readout
    ,output empty
    ,output full
    ,output dout[WID-1:0]
);
```