

# Specification

## Synchronous FIFO

Nebhrajani A.V.

July 13, 2022

### 1 Operation

The high-level operation of a synchronous FIFO can be modeled by a circular buffer as shown:

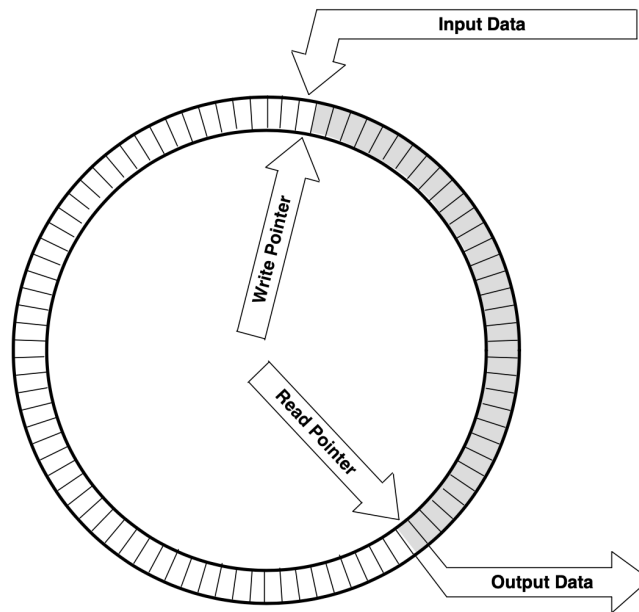


Figure 1: Circular buffer modelling a FIFO.

Two operations are defined: reading and writing to the FIFO.

Writing input data involves:

- Copying the input data to the location `write pointer` points to.
- Incrementing `write pointer`.

Reading data involves:

- Reading out the data at location `read pointer`.
- Incrementing `read pointer`.

“Synchronous” refers to the fact that there is a single common free-running clock for both reading and writing. When to read and write is given by two signals: a read enable and a write enable. Our FIFO has 64 addresses and each location stores an 8-bit word (one byte).

## 2 Interface

### 2.1 Description

Signal Name	Width	Type	Description	Drive
<code>rst</code>	1	Input	Reset device	Active low
<code>clk</code>	1	Input	Free running clock	Square wave
<code>w_en</code>	1	Input	Write enable	Active high
<code>din</code>	8	Input	Input for WRITE operation.	Active high
<code>r_en</code>	1	Input	Read enable	Active high
<code>dout</code>	8	Output	Output of READ operation.	Active high
<code>full</code>	1	Output	Raised when FIFO is full.	Active high
<code>empty</code>	1	Output	Raised when FIFO is empty/on reset.	Active high
<code>overflow</code>	1	Output	Raised if write is attempted when <code>full</code> .	Active high
<code>underflow</code>	1	Output	Raised if read is attempted when <code>empty</code> .	Active high

### 2.2 Verilog

```
module sfifo
(
    rst,
    clk,
    w_en,
    din,
    r_en,
    dout,
    full,
    empty,
    overflow,
    underflow
);
```

```

input          rst;
input          clk;
input          w_en;
input  [7:0]   din;
input          r_en;
output  [7:0]   dout;
output         full;
output         empty;
output         overflow;
output         underflow;

endmodule

```

### 3 Timing Diagrams

