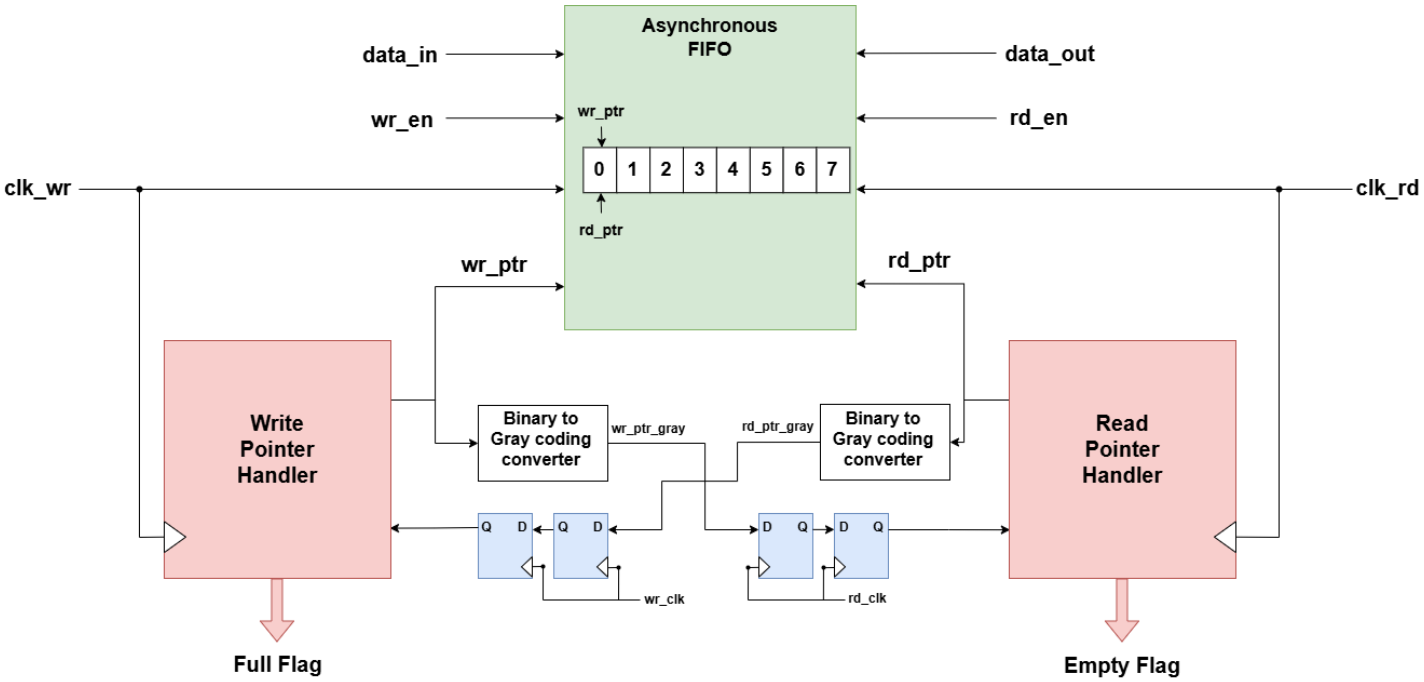
Asynchronous FIFO

➢ Introduction

- In an asynchronous FIFO, the read and write operations run on different clock frequencies. Because the write and read clocks are not synchronized, the FIFO is termed "asynchronous." It is commonly used for transferring data between two different clock domains, a process known as clock domain crossing (CDC). This allows smooth synchronization of data flow between systems operating on independent clocks.
- In a synchronous FIFO, both write and read pointers are driven by the same clock. In contrast, an asynchronous FIFO uses a write pointer tied to the write clock domain and a read pointer tied to the read clock domain. As a result, domain crossing is necessary to determine FIFO full and empty conditions, which can introduce metastability into the design. To address this issue, 2-flip-flop synchronizer is used to transfer the write and read pointers across domains.

➢ System Architecture



✓ It's important to note that a single 2-FF synchronizer can resolve metastability for only single bit, so if we use multiple 2-FF synchronizers for write and read pointers → **Data Incoherency issue** will be arisen. Therefore, to address this issue we use **Binary to Gray Converter** for both read and write pointers before crossing the second domain.

➢ Parameters and Ports Declaration

Signal Name	Width	Direction	Description
FIFO_WIDTH	-	Parameters	FIFO word width
FIFO_DEPTH	-		Number of FIFO locations
ADDR_SIZE	-		Address line of FIFO
data_in	FIFO_WIDTH	Inputs	Data written in the FIFO at the write domain in case of write operation
wr_en	1		Write Enable: If the FIFO is not full, asserting this signal causes $data\_in$ to be written into the FIFO
rd_en	1		Read Enable: If the FIFO is not empty, asserting this signal causes $data\_out$ to be read from the FIFO
clk_wr	1		Clock signal of the write domain
clk_rd	1		Clock signal of the read domain
rst_n	1		Active low Asynchronous system reset in both domains
data_out	FIFO_WIDTH	Outputs	Read data from the FIFO at the read domain in case of read operation
full	1		Flag Indicates that the FIFO is full (there is no space available to write new data)
empty	1		Flag Indicates that the FIFO is empty (there is no data available to read)
overflow	1		Flag indicates that the FIFO is full & the write enable is also activated
underflow	1		Flag indicates that the FIFO is empty & the read enable is also activated

- Notice that the default values of the above parameters are:
  1) **FIFO_WIDTH** = 4 bits
  2) **FIFO_DEPTH** = 8 locations
  3) **ADDR_SIZE** = $\log_2(8) = 3\ bits$