



BÀI TẬP LỚN THIẾT KẾ LUẬN LÝ(CO1026)

FIFO (First in First Out)

GVHD: Nguyễn Thiên Ân
Lớp L05 - Nhóm 06

Danh sách thành viên

MSSV	Sinh viên thực hiện
2112173	Phạm Hồng My Sa
2213857	Võ Ngọc Tú
2211186	Lê Đăng Huy
2213561	Trịnh Thu Trang

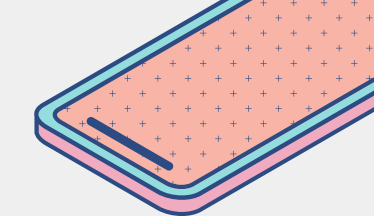
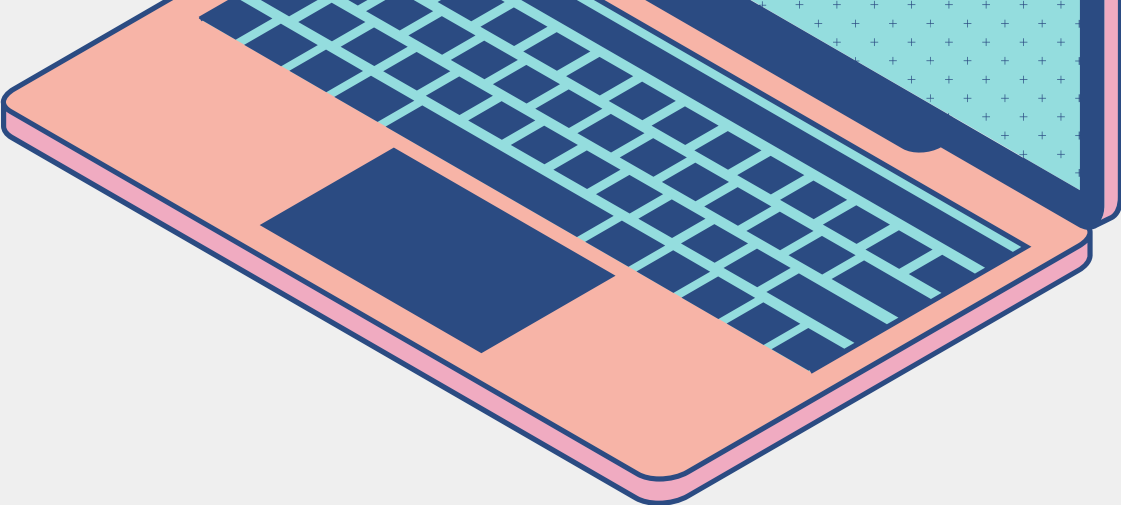


Table of contents

1

2

3

4

5

PHẦN

PHẦN

PHẦN

PHẦN

PHẦN

Khái niệm và
ứng dụng của
First In First
Out(FIFO)

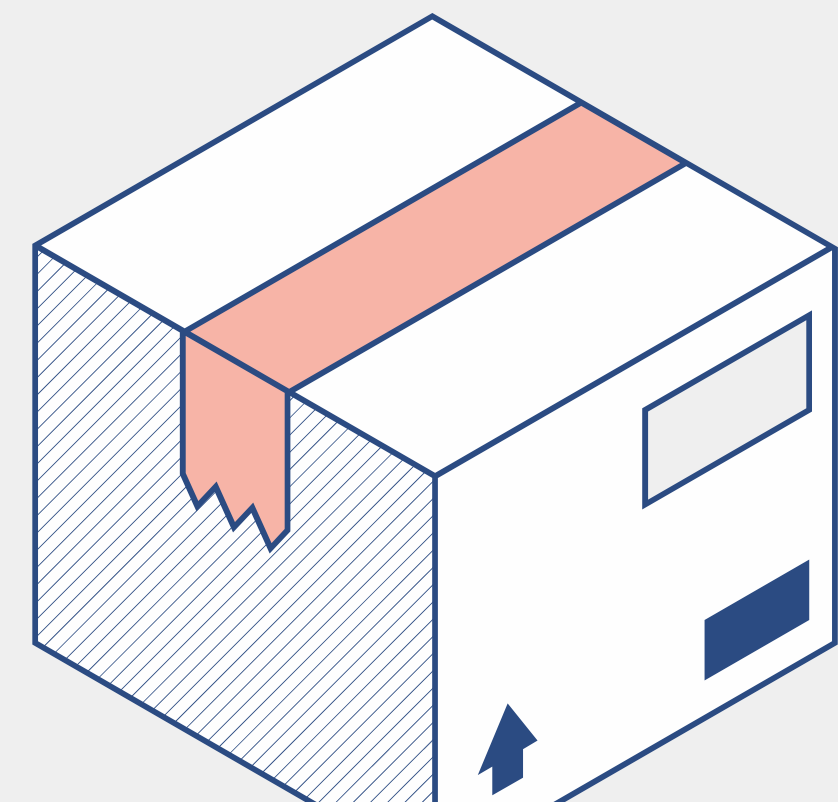
Thiết kế và xây
dựng mạch
FIFO sử dụng
Verilog HDL

Testbench


Demo on board

Tài liệu tham
khảo

1. Khái niệm
2. Hiện thực
3. Ứng dụng



Giới thiệu về đề tài



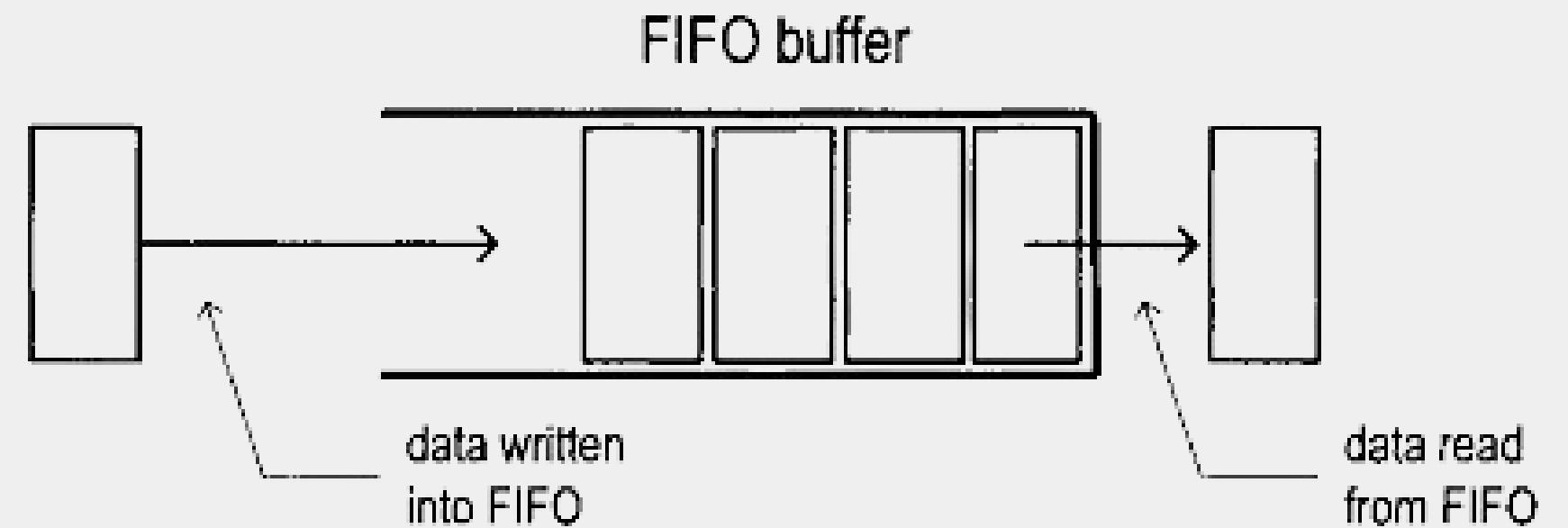
Trong các thiết bị kỹ thuật, dữ liệu sẽ được chuyển đổi giữa các bảng mạch in, việc sử dụng bộ lưu trữ trung gian hoặc bộ nhớ đệm là cần thiết khi dữ liệu đến bảng mạch in nhận với tốc độ cao hoặc theo từng đợt, nhưng hạn chế là xử lý không được mượt mà, tốc độ xử lý không nhanh.

Ta dễ dàng bắt gặp được các bộ nhớ đệm trong cuộc sống hằng ngày. VD: hàng đợi khách tại điểm thanh toán trong siêu thị, hàng xe ô tô di chuyển tại đèn giao thông,...

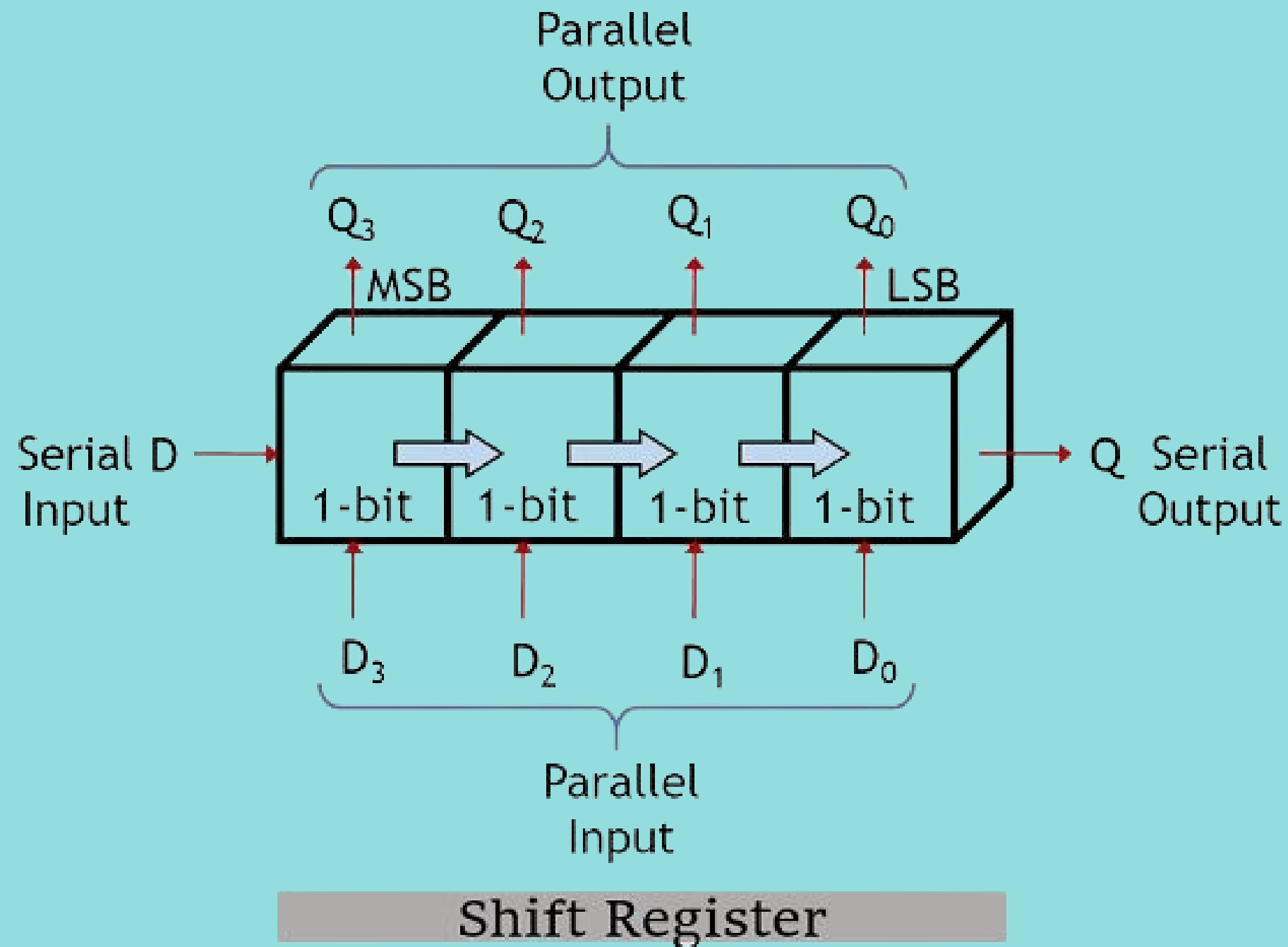
Điểm thanh toán làm việc tuần tự, chậm rãi trong khi lượng khách hàng không ổn định → Khái niệm First In First Out (ai tới trước sẽ phục vụ trước)

Phần 1: Khái niệm và ứng dụng First In First Out (FIFO)

1. Khái niệm:



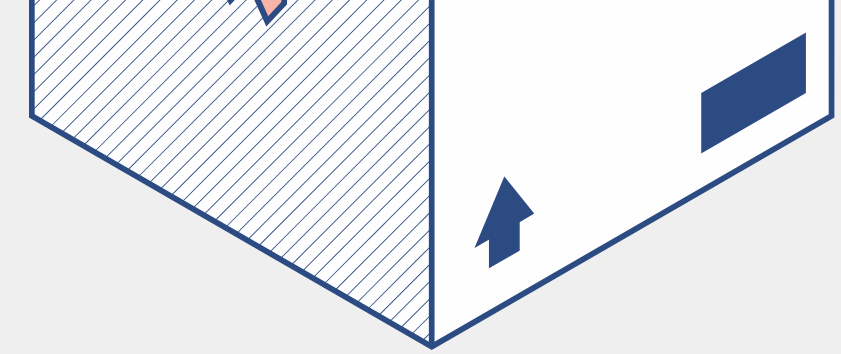
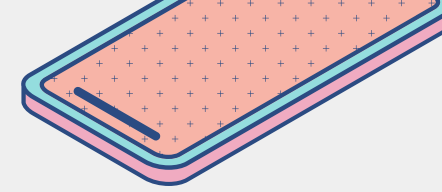
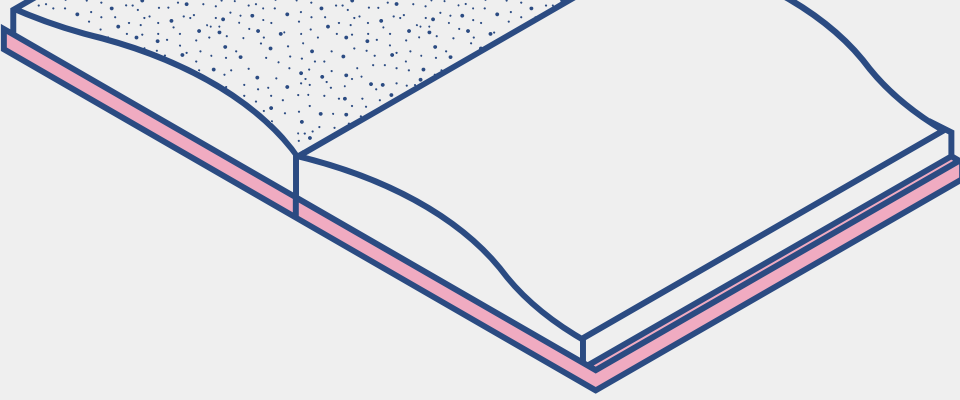
- FIFO là một bộ đệm đặc biệt(First In First Out), dữ liệu nào được ghi vào bộ đệm trước sẽ ra khỏi bộ đệm trước. FIFO có thể hiện thực bằng phần mềm hay phần cứng phụ thuộc vào ứng dụng đang được sử dụng và các tính năng mong muốn.
- Phần mềm linh hoạt hơn, nhưng bù lại lợi ích của phần cứng nằm ở tốc độ của nó.
- Bất kì bộ nhớ nào mà dữ liệu được viết vào đầu tiên cũng sẽ ra đầu tiên khi dữ liệu được đọc là bộ nhớ FIFO. Có 3 loại FIFO: *Shift register FIFO*, *Exclusive read/write FIFO*, *concurrent read/write FIFO*.



a. Shift register (thanh ghi dịch)

Thanh ghi dịch FIFO trong đó số lượng dữ liệu có thể được phép lưu trữ là cố định, để đảm bảo sự đồng bộ giữa thao tác đọc và ghi, cần phải đọc mỗi khi một dữ liệu khác được ghi vào.

Về cơ bản, thanh ghi dịch là một mạch FIFO đơn hướng, nó dịch từng bit đơn của dữ liệu được đưa vào input ra output trong mỗi chu kỳ clock



b. Exclusive read/write FIFO

- FIFO chỉ đọc/ghi với số lượng dữ liệu được lưu trữ không cố định
- Trong cơ chế này cần có sự đồng bộ giữa các thao tác đọc và ghi, có mối quan hệ thời gian giữa write clock và read clock [không được chồng chéo (overlap) read clock và write clock]



c. Concurrent read/write FIFO

- FIFO đọc và ghi đồng thời với số lượng dữ liệu được lưu trữ không cố định và có sự không đồng bộ giữa thao tác đọc và ghi.
- Trong cơ chế này, ta có thể đồng thời đọc và ghi chồng chéo nhau do không phụ thuộc giữa đọc và ghi dữ liệu.
- Do đó các tín hiệu điều khiển cho việc đọc và ghi được phân thành 2 nhóm: FIFO đồng bộ (Synchronous FIFO) và FIFO bất đồng bộ (Asynchronous FIFO).

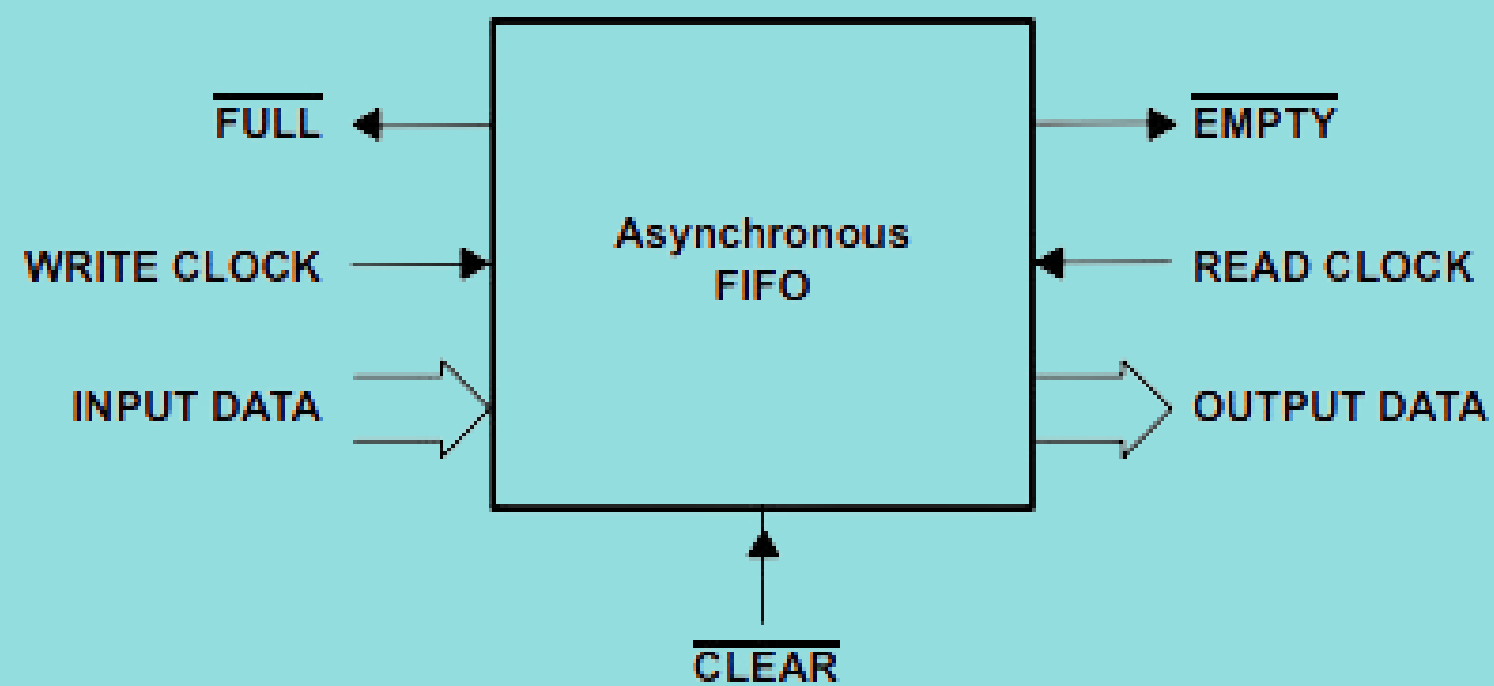


Figure 8. Connections of an Asynchronous FIFO

HÌNH ẢNH MINH HỌA CHO
ASYNCHRONOUS FIFO

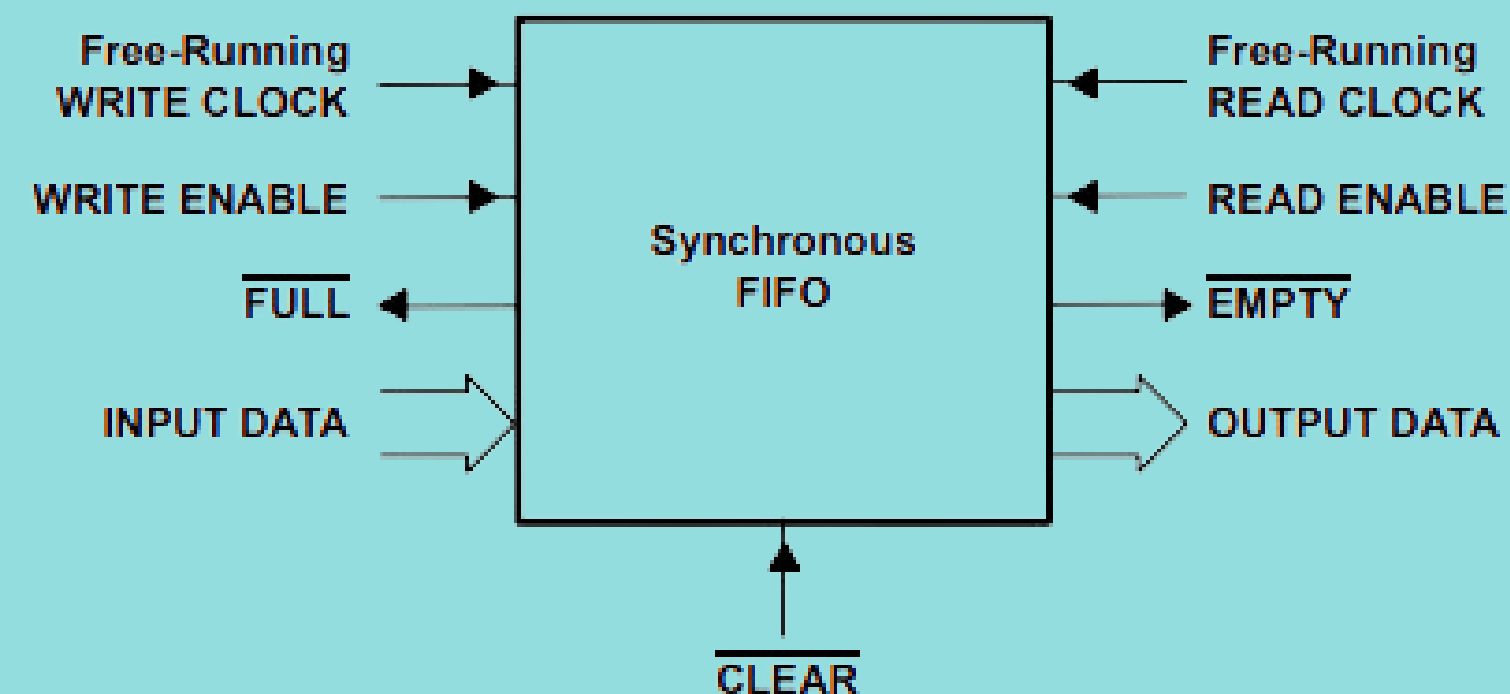
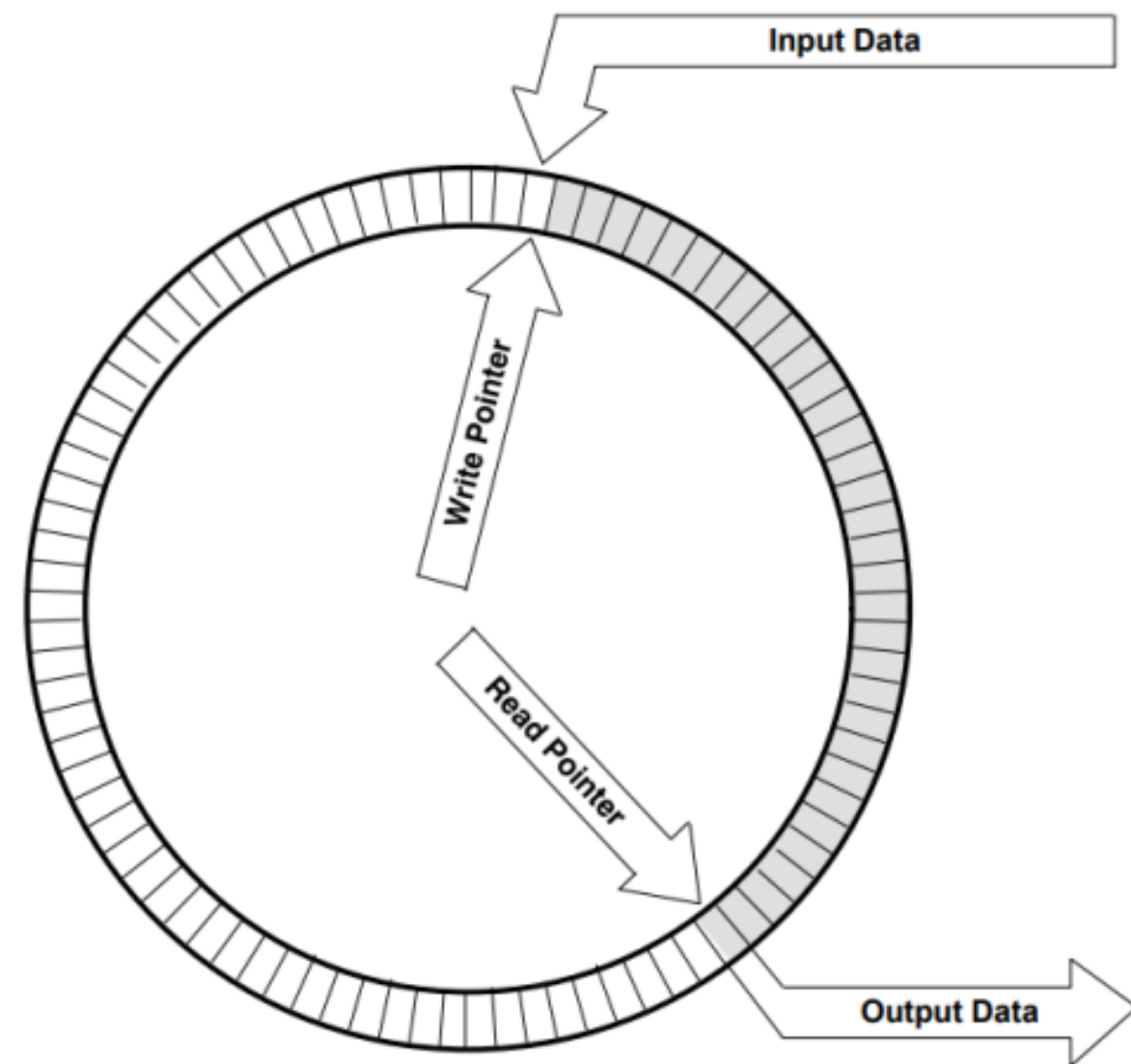


Figure 11. Connections of a Synchronous FIFO

HÌNH ẢNH MINH HỌA
SYNCHRONOUS FIFO

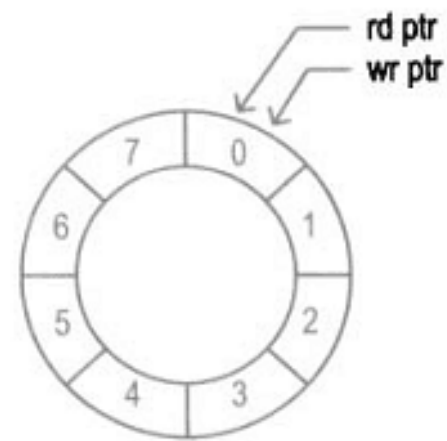
Trong bài báo cáo này, nhóm sẽ nói về việc thiết kế và hiện thực FIFO đồng bộ. Mọi hệ thống xử lý kỹ thuật số đều hoạt động đồng bộ với tín hiệu clock trên toàn hệ thống, và hệ thống sẽ tiếp tục chạy ngay cả khi không có hành động nào được thực thi. Tín hiệu kích hoạt(enable signals) bao gồm write enable, read enable, còn được gọi là tín hiệu chọn chip, được sử dụng để đồng bộ hóa các thao tác và đọc trong các thiết bị khác nhau như bộ nhớ và cổng.

2. Hiện thực

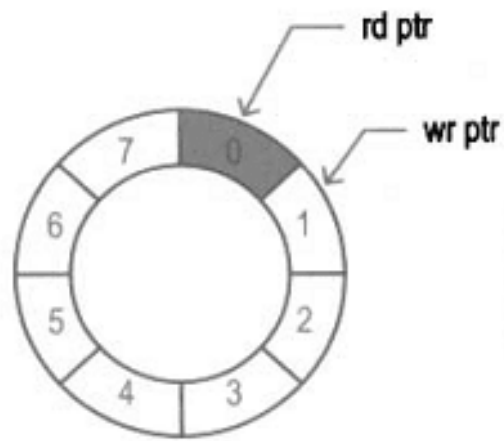


Minh họa cho ring buffer

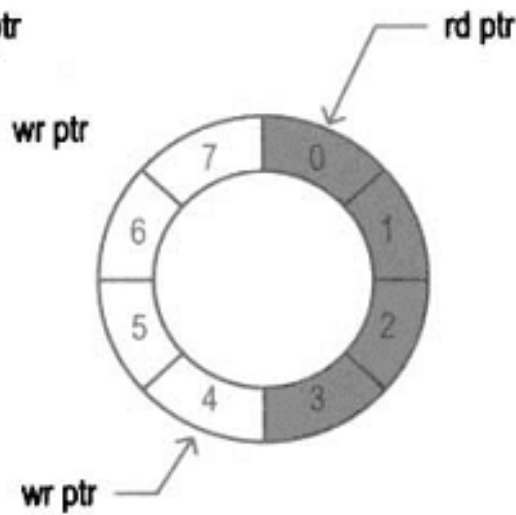
- FIFO có thể hiện thực bằng thanh ghi dịch - phần cứng (hardware shift register) hoặc dùng các cấu trúc bộ nhớ khác, điển hình là FIFO vòng (circular buffer, ring buffer, array - base buffer).
- Trong khoa học máy tính, buffer vòng là một cấu trúc dữ liệu được sử dụng một buffer đơn, có kích thước cố định và được kết nối đầu cuối với nhau (end - to - end).



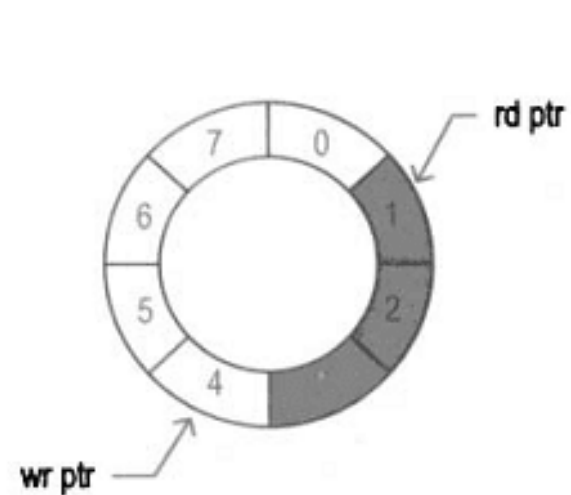
(a). initial (empty)



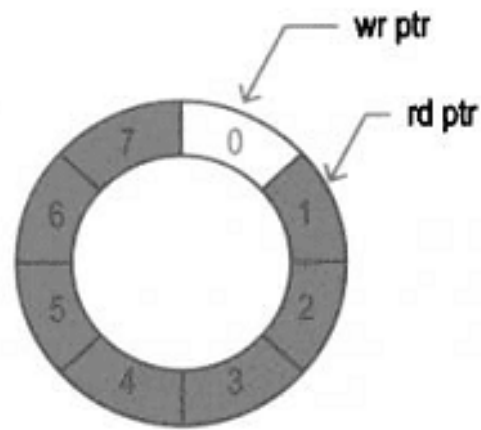
(b). after a write



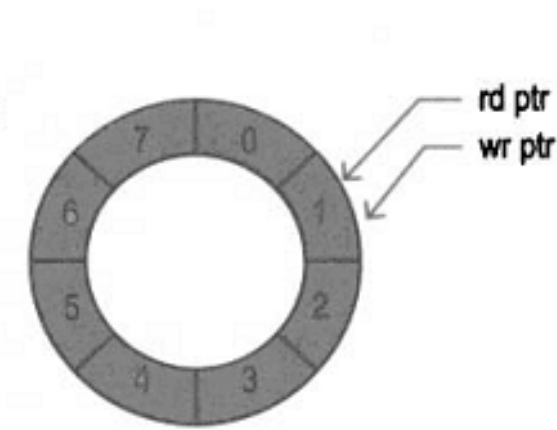
(c). 3 more writes



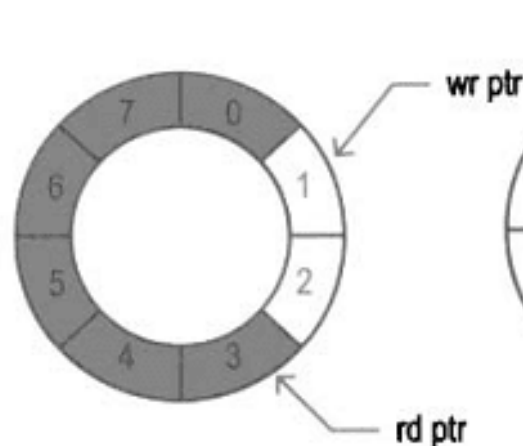
(d). after a read



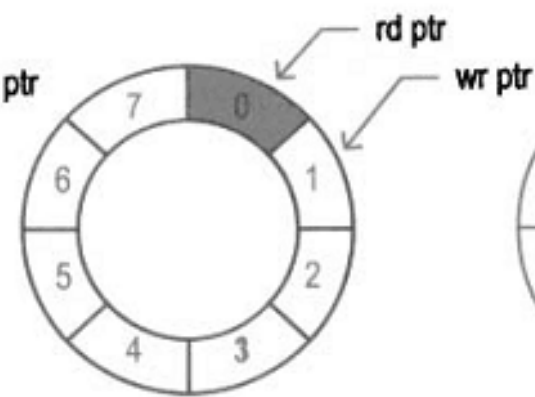
(e). 4 more writes



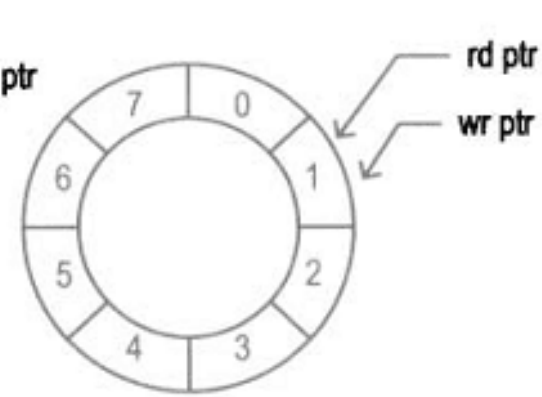
(f). 1 more write (full)



(g). 2 reads



(h). 5 more reads

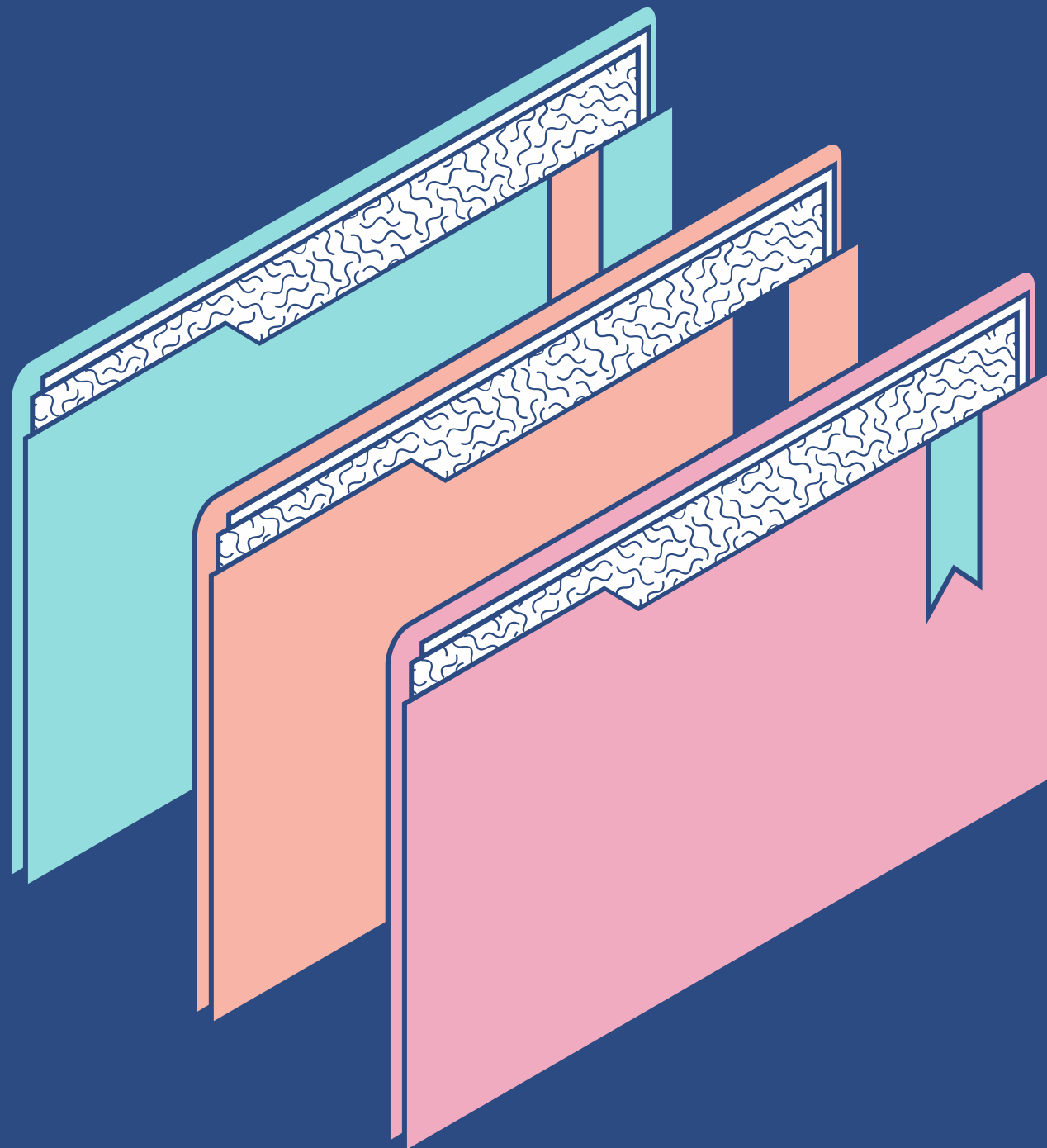


(i). 1 more read (empty)

Hiện thực FIFO bằng buffer vòng: Như tên gọi của nó (array - base), bộ đệm này được thực hiện dựa trên một mảng. Kèm theo đó là 2 con trỏ Write và Read.

- Khi bắt đầu, con trỏ rd ptr và wr ptr đều cùng trỏ vào vị trí đầu tiên (index 0).
- Khi nhận lệnh ghi, wr ptr sẽ ghi data vào bộ đệm, sau đó tăng lên 1 đơn vị. Khi nhận lệnh đọc, dữ liệu được rd ptr trỏ vào sẽ được đọc ra và rd ptr tăng lên 1 đơn vị.
- Khi 1 con trỏ tới cuối mảng, nó cuộc lại vị trí đầu tiên. Đó là lý do vì sao gọi đây là bộ đệm vòng.
- $(wr\ ptr + 1) == rd\ ptr \rightarrow$ bộ nhớ đệm đầy (full)
- $(wr\ ptr == rd\ ptr) \rightarrow$ bộ nhớ đệm trống (empty)

3. Ứng dụng



a. FIFO thường được sử dụng trong các mạch điện tử để đệm (buffering) và điều khiển luồng (flow) giữa phần cứng và phần mềm.

b. FIFO cũng thường được ứng dụng trong lĩnh vực quản lý hàng hóa. Phương pháp FIFO là cách quản lý kho theo hình thức nhập trước xuất trước.

- Ưu điểm :

- Hàng hóa được lưu kho với thời gian ngắn → chi phí tồn kho của sản phẩm giảm → giúp tăng lợi nhuận doanh nghiệp
- Với đặc thù luân chuyển hàng hóa liên tục, doanh nghiệp có thể kiểm soát chất lượng sản phẩm tốt hơn

- Nhược điểm :

- Cần không gian lưu trữ kho lớn với nhiều thiết bị chuyên dụng
- Cần hệ thống phần mềm theo dõi hàng tồn kho một cách kỹ càng, tránh bỏ sót hàng hóa.



Phần 2: Thiết kế và xây dựng mạch FIFO sử dụng Verilog HDL

- Dựa vào ý tưởng trình bày trên, nhóm đã nghiên cứu và đưa ra đoạn code hoàn chỉnh cho một FIFO cơ bản được thực hiện bằng cấu trúc ring buffer.

Source code : BTL.v

```
module BTL(  
    input wire clk, // clock  
    input wire rst, // reset  
    input wire wr_en, // write enable  
    input wire rd_en, // read enable  
    input wire [7:0] data_in, // input data  
    output reg [7:0] data_out, // output data  
    output reg empty, // FIFO empty flag  
    output reg full // FIFO full flag  
);
```

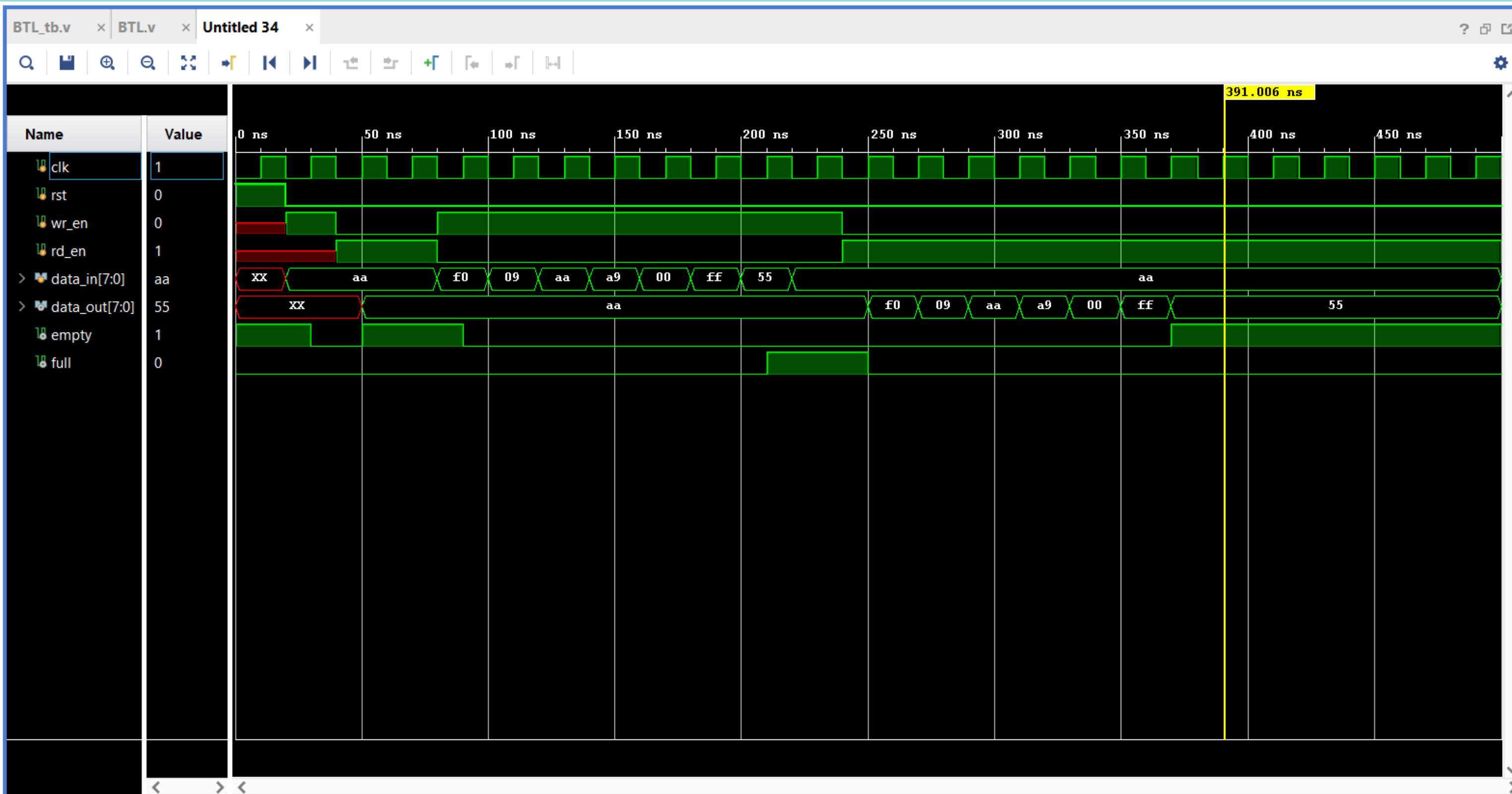
Các thành phần trong module

```
// Define the FIFO depth (FIFO buffer size)  
parameter DEPTH = 8;  
  
// Create a register to hold the FIFO data (buffer can store 8 data 8bit-words)  
reg [7:0] mem [0 : DEPTH - 1];  
  
// wr_ptr / rd_ptr  
reg [2:0] rd_idx = 0;  
reg [2:0] wr_idx = 0;  
  
// Initialize FIFO empty and full flags  
always @(*) begin  
    empty = (wr_idx == rd_idx);  
    full = ((wr_idx + 1) % DEPTH == rd_idx);  
end
```

```
// Write operation
always @(posedge clk) begin
    if (rst) begin
        wr_idx <= 0;
    end
    else if (wr_en && !full) begin
        mem[wr_idx] <= data_in;
        wr_idx <= (wr_idx + 1) % DEPTH;
    end
end

// Read operation
always @(posedge clk) begin
    if (rst) begin
        rd_idx <= 0;
    end
    else if (rd_en && !empty) begin
        data_out <= mem[rd_idx];
        rd_idx <= (rd_idx + 1) % DEPTH;
    end
end
```

Phần 3: Testbench



Source code:
BTL_tb.v

Demo on board

Source code: BTL_board.v

Code BTL_board.v dùng để hiện thực trên board thay đổi so với code BTL.v ở hai nội dung: (1) thay đổi kích thước dữ liệu data_in vào FIFO từ 8 bit thành 1 bit để đơn giản hơn trong việc hiện thực trên board để quan sát data_out (1 bit: bit 0 - đèn tắt, bit 1 - đèn sáng) và (2) clk được scale lại (thông qua temp_clk với chu kỳ dài hơn) để có thể dễ dàng quan sát output.



Phần 5: Tài liệu tham khảo



1. <https://hgn37.wordpress.com/2017/05/20/bo-dem-fifo-vong/>

2. https://en.wikipedia.org/wiki/Circular_buffer

3. <https://www.ti.com/lit/an/scaa042a/scaa042a.pdf>

4.

[https://en.wikipedia.org/wiki/FIFO_\(computing_and_electronics\)](https://en.wikipedia.org/wiki/FIFO_(computing_and_electronics))

5. https://en.wikipedia.org/wiki/Queueing_theory

6. <https://vimach.net/threads/thiet-ke-bo-nho-dem-fifo-dung-verilog.124/>

7. <https://itgtechnology.vn/quan-ly-kho-fifo-lifo/>

8. <https://als.com.vn/phuong-phap-fifo-first-in-first-out-trong-xuat-nhap-hang>



Thank you

CẢM ƠN CÁC BẠN VÀ THẦY ĐÃ LẮNG NGHE.

NHÓM ĐÃ NGHIÊN CỨU VÀ HOÀN THÀNH ĐỀ TÀI NHƯNG VẪN CÒN NHIỀU THIẾU SÓT, HI VỌNG NHẬN ĐƯỢC SỰ QUAN TÂM VÀ GÓP Ý TỪ GIẢNG VIÊN CŨNG NHƯ CÁC BẠN ĐỂ CẢI THIỆN HƠN.

