

3.3. Mô tả các phần tử nhớ Latch, Flip-flop theo sườn và theo mức (Mô tả phần tử chốt theo mức dùng phép gán liên tục; Mô tả phần tử Flip-flop theo sườn dùng cấu trúc always và phép gán non-blocking; Khái niệm về sườn đồng hồ; Mô tả tín hiệu khởi tạo Flip-flop không đồng bộ; So sánh phép gán non-blocking và phép gán blocking)-2 LT

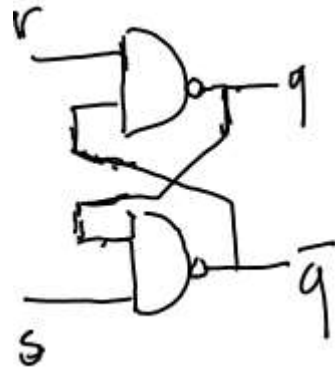
### 3.3.1. Mô tả phần tử chốt (latch)

- Sử dụng phép gán liên tục với biến được gán giá trị ở phía trái phép gán được sử dụng ở biểu thức phía phải phép gán
  - Ví dụ: `assign q_latch = (en==1)?d_in:q_latch;`
  - Chú ý: Sự phụ thuộc vòng giữa các biến trong phép gán liên tục cũng tạo ra latch, nhưng nên hạn chế dùng

- Ví dụ

`assign q_rs = ~(r & q_bar_rs);`

`assign q_bar_rs = ~(s & q_rs);`



### 3.3.2. Mô tả phần tử flip-flop

- Sử dụng khối always với danh sách độ nhạy là sườn xung nhịp

- Cú pháp

**always @(posedge clk)**

**begin**

....các\_lệnh...

**end**

**always @(negedge clk)**

**begin**

....các\_lệnh...

**end**

- Hoạt động: các\_lệnh sẽ được thực hiện khi có sườn lên (xuống) của xung nhịp đồng hồ
- Tổng hợp: thành các flip-flop điều khiển bởi sườn lên (xuống) của xung nhịp
- Các phép gán trong khối always điều khiển bằng sườn đồng hồ thường là các phép gán song song (`<=`) (non-blocking)
- Các câu lệnh lựa chọn (if, case) trong khối always điều khiển bằng sườn đồng hồ không cần có đầy đủ các nhánh vì luôn tạo ra flip-flop
- Ví dụ:

**module shifter (**

**input d\_in,**

**input clk,**

**output d\_out);**

**reg a, b, c;**

```

always @(posedge clk)
begin
    a <= d_in;
    b <= a;
    c <= b;
    d_out <= c;
end
endmodule

```

- Phân biệt giữa phép gán song song và phép gán tuần tự  
Tham khảo: Clifford E. Cummings, "Nonblocking Assignments in Verilog Synthesis, Coding Styles That Kill!"

### 3.3.3. Mô tả phần tử flip-flop có tín hiệu reset

- Reset đồng bộ: tín hiệu reset không nằm trong danh sách điều khiển khối **always**

```

always @(posedge clk)
begin
    if (reset)
        q <= 1'b0;
    else
        q <= d;
end

```

- Reset không đồng bộ: tín hiệu reset (sườn của nó) nằm trong danh sách điều khiển khối **always**

```

always @(posedge clk or posedge reset)
begin
    if (reset)
        q <= 1'b0;
    else
        q <= d;
end

```

	posedge clk	negedge clk
reset đồng bộ mức 0		
reset đồng bộ mức 1		
reset không đồng bộ, sườn âm		
reset không đồng bộ, sườn dương		

- Chú ý: Khi mô tả flip-flop cần chú ý tới các phần tử có trong thư viện chuẩn. Phần mềm tổng hợp sử dụng các phần tử có sẵn => một số loại flip-flop sẽ cần 2 phần tử (hoặc nhiều hơn) từ thư viện chuẩn => có kích thước lớn hơn. Nên sử dụng loại flip-flop có hỗ trợ trong thư viện chuẩn (kích thước mạch sẽ nhỏ hơn).

