

3.2.2. Mô hình hành vi

- a) Khối lệnh always
- b) Câu lệnh gán tuần tự (blocking)
- c) Câu lệnh điều kiện if/else
- d) Câu lệnh lựa chọn case
- e) Câu lệnh lặp
 - Cú pháp
 - **for** (lệnh_khởi_tạo; điều_kiện; lệnh_chỉ_số) **begin ... end**
 - **repeat** (số_lần_lặp) **begin ... end**
 - **while** (điều_kiện) **begin ... end**
 - Hoạt động
 - for:
 - Khi bắt đầu vòng lặp, lệnh_khởi_tạo được thực hiện để gán giá trị bắt đầu cho biến chỉ số điều khiển vòng lặp
 - Thực hiện lặp các lệnh:
 - Kiểm tra biểu thức điều_kiện, nếu biểu thức sai thì kết thúc lặp
 - Nếu biểu thức điều_kiện đúng thì thực hiện các lệnh trong vòng lặp bao bởi begin ... end
 - Thực hiện lệnh_chỉ_số để thay đổi giá trị biến chỉ số
 - repeat: Thực hiện lặp số_lần_lặp lần các lệnh bao giữa begin...end
 - while: Thực hiện các lệnh bao giữa begin...end đến khi nào biểu thức điều_kiện sai.
 - Tổng hợp
 - Với các lệnh lặp có số lần lặp cố định (vòng lặp tĩnh - static loop), phần mềm tổng hợp sẽ trải vòng lặp thành khối lệnh thủ tục thông thường và thực hiện tổng hợp
 - Với các lệnh lặp có số lần lặp phụ thuộc biến (vòng lặp động non-static loop): phần mềm tổng hợp sẽ thường không tổng hợp thành mạch được. Tuy nhiên có một số trường hợp đặc biệt với phần mềm tổng hợp mức cao (high level synthesis), vòng lặp động có thể tổng hợp được.
 - Ví dụ: Mạch majority

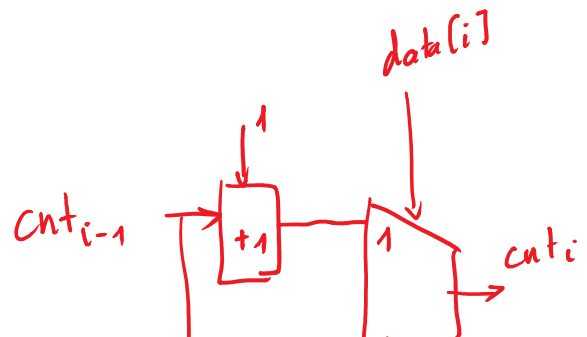
module majority

```
# (parameter data_width = 8,
  cnt_width = 4,
  majority_value = 4
)
(
  input [data_width-1:0] data,
  output reg y
);
```

```
reg [cnt_width-1:0] cnt;
integer i;
```

```
always @(data)
begin
```

```
  cnt = 0;
  for (i = 0; i < data_width; i = i + 1)
  begin
    if (data[i]) cnt = cnt + 1; ✗
```



```

    for (i = 0; i < data_width; i = i + 1)
    begin
        if (data[i]) cnt = cnt + 1;
    end
    if (cnt > majority_value)
        y = 1;
    else
        y = 0;
    end
endmodule

```

```

module test();

```

```

    reg [7:0] data;
    wire      y;

```

```

    majority_for duv(.data(data), .y(y));

```

```

    initial

```

```

        begin

```

```

            $monitor ("%t: data=%b, y=%b", $time, data, y);

```

```

            repeat (20)

```

```

                begin

```

```

                    data = $random();

```

```

                    #5;

```

```

                end

```

```

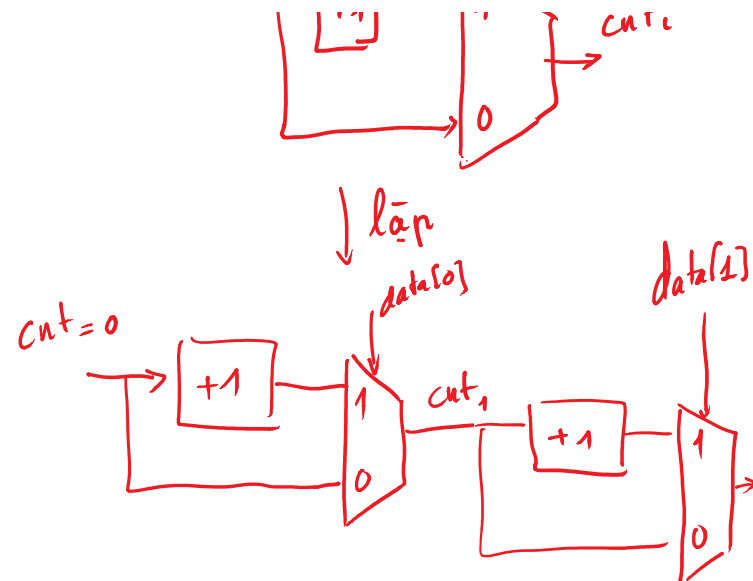
            end

```

```

        endmodule // test

```

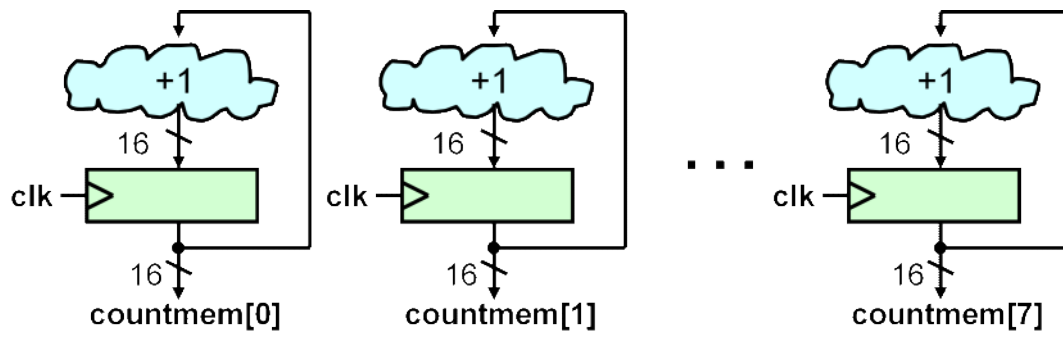


Ví dụ vòng for trong khối always có đồng hồ

```

reg [15:0] countmem [0:7];
integer x;
always @(posedge clk) begin
    for (x = 0; x < 8; x = x + 1) begin
        countmem[x] <= countmem[x] + 1;
    end
end

```



a) Chương trình con

- Chương trình con được sử dụng để đóng gói một đoạn mã và sử dụng lại đoạn mã đó nhiều lần.