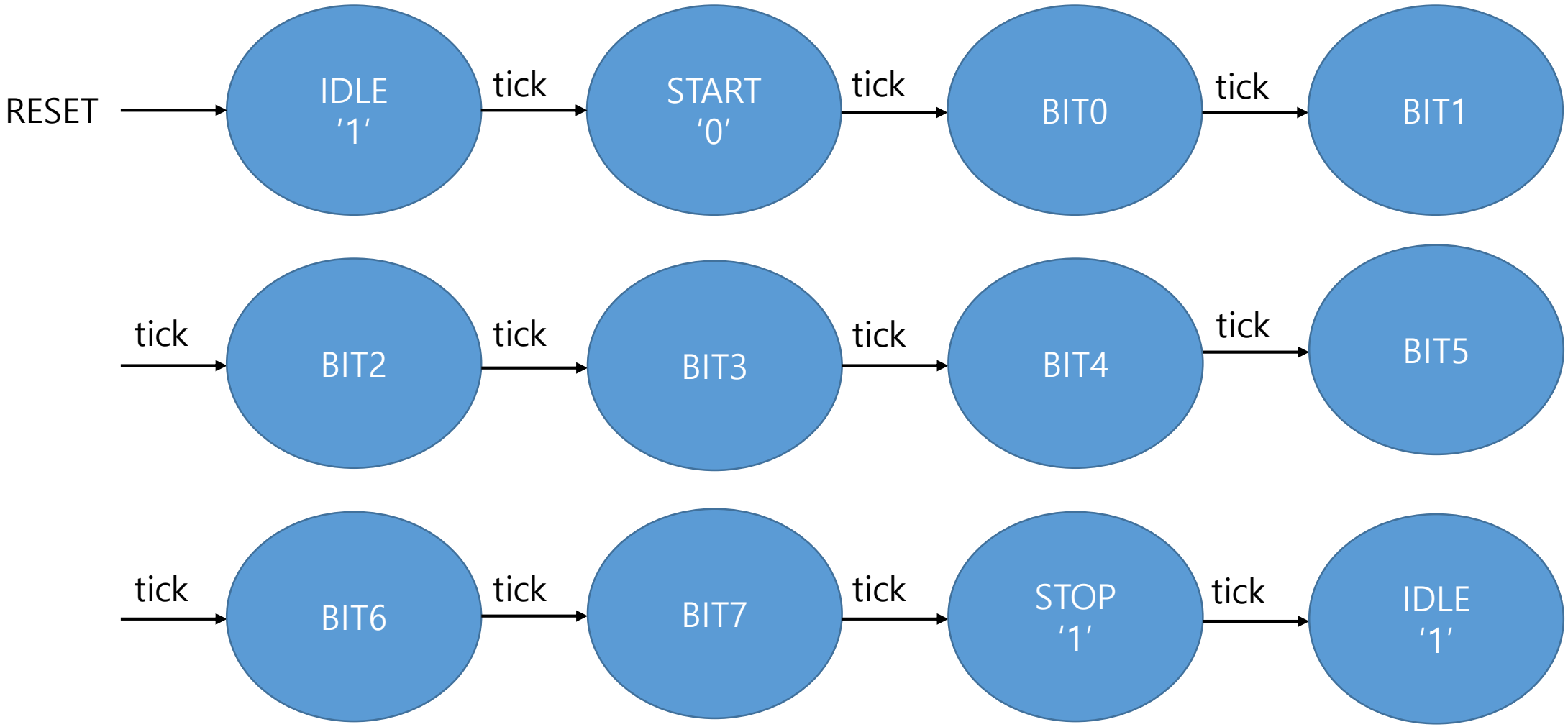
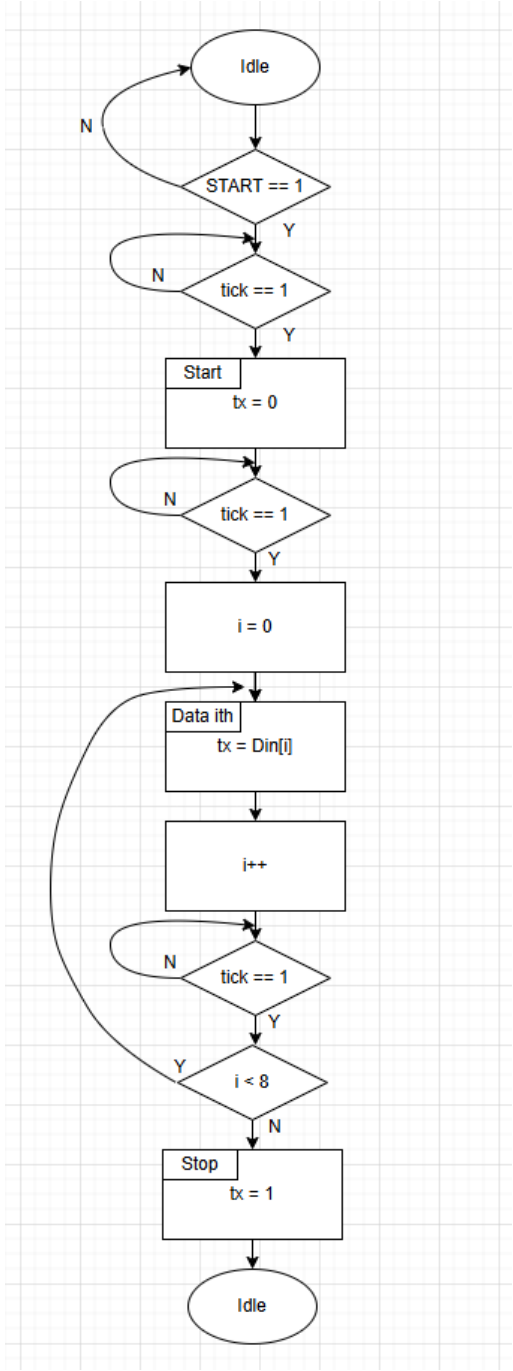
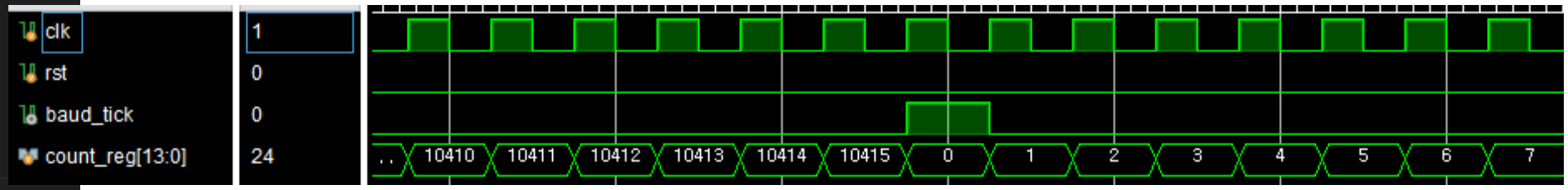

RESET
IDLE
'1'
tick
START
'0'
tick
BIT0
tick
BIT1
tick
BIT2
tick
BIT3
tick
BIT4
tick
BIT5
tick
BIT6
tick
BIT7
tick
STOP
'1'
tick
IDLE
'1'

ASM

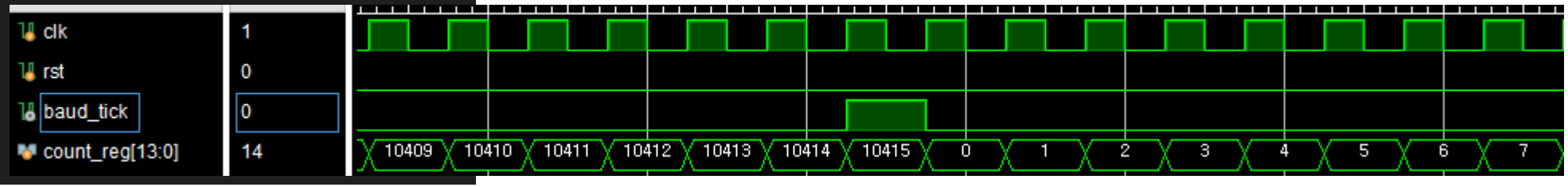
Transfer one byte without parity bit



```
assign baud_tick = baud_tick_reg;

always @(posedge clk, posedge rst) begin
    if(rst) begin
        count_reg <= 0;
        baud_tick_reg <= 1'b0;
    end else begin
        count_reg <= count_next;
        baud_tick_reg <= baud_tick_next;
    end
end

always @(*) begin
    count_next = count_reg;
    baud_tick_next = 1'b0; // baud_tick_reg여도 상관 없음

    if(count_reg == BAUD_COUNT - 1) begin
        count_next = 0;
        baud_tick_next = 1'b1;
    end else begin
        count_next = count_reg + 1;
        baud_tick_next = 1'b0;
    end
end
```




```
assign count_next = (count_reg == BAUD_COUNT - 1) ? 0 : count_reg + 1;
assign baud_tick = (count_reg == BAUD_COUNT - 1) ? 1'b1 : 1'b0;

always @(posedge clk, posedge rst) begin
    if(rst) begin
        count_reg <= 0;
    end else begin
        count_reg <= count_next;
    end
end
```



차이점: baud_tick의 발생 시점이 서로 다름.
이유: assign으로 짜여진 구조는 바로 baud_tick을 생성하는 반면,
always(*)로 짜여진 구조는 baud_tick_next를 한번 거치기 때문에 한 클럭이 지난 후 baud_tick이 생성됨.