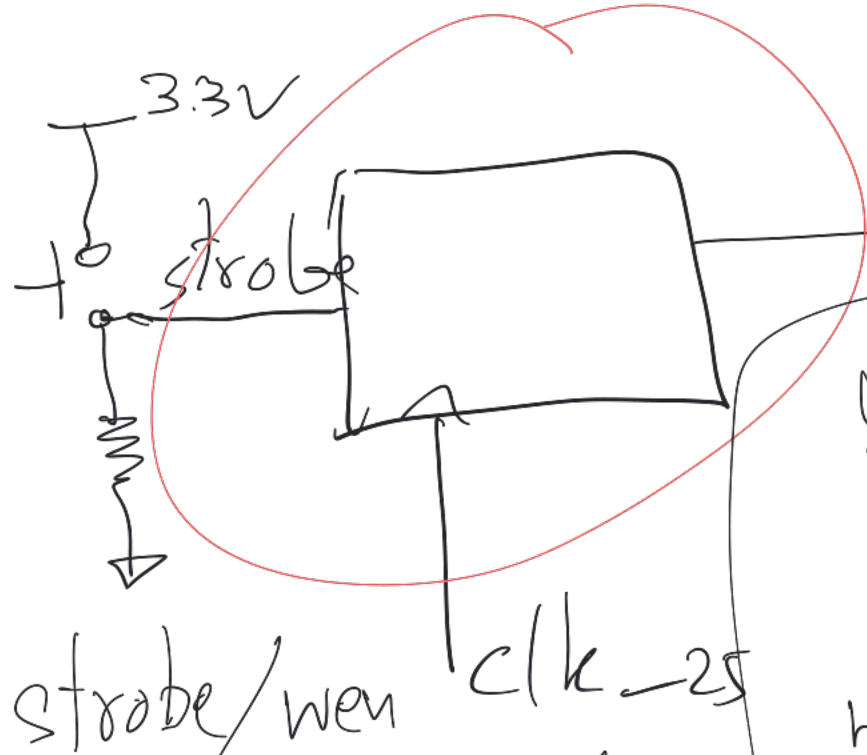
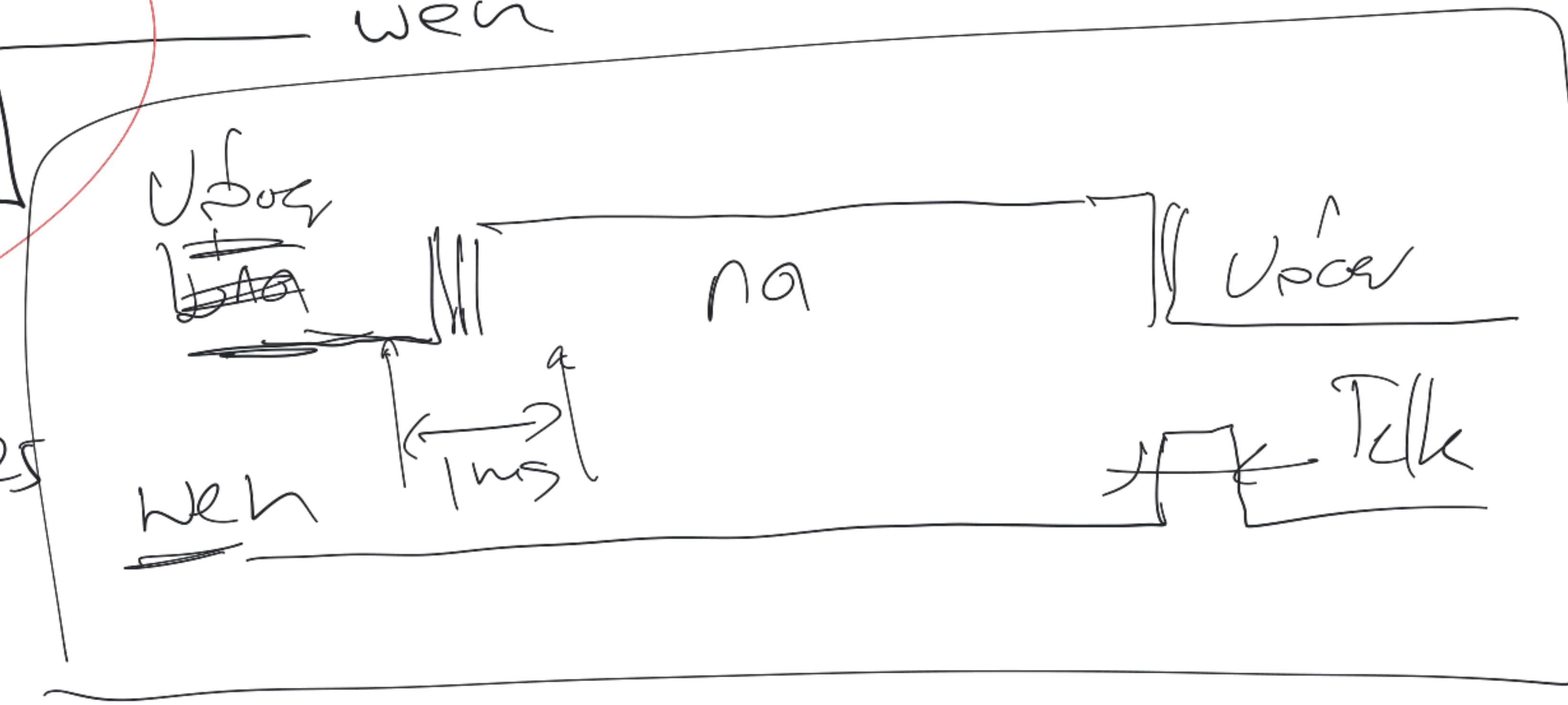
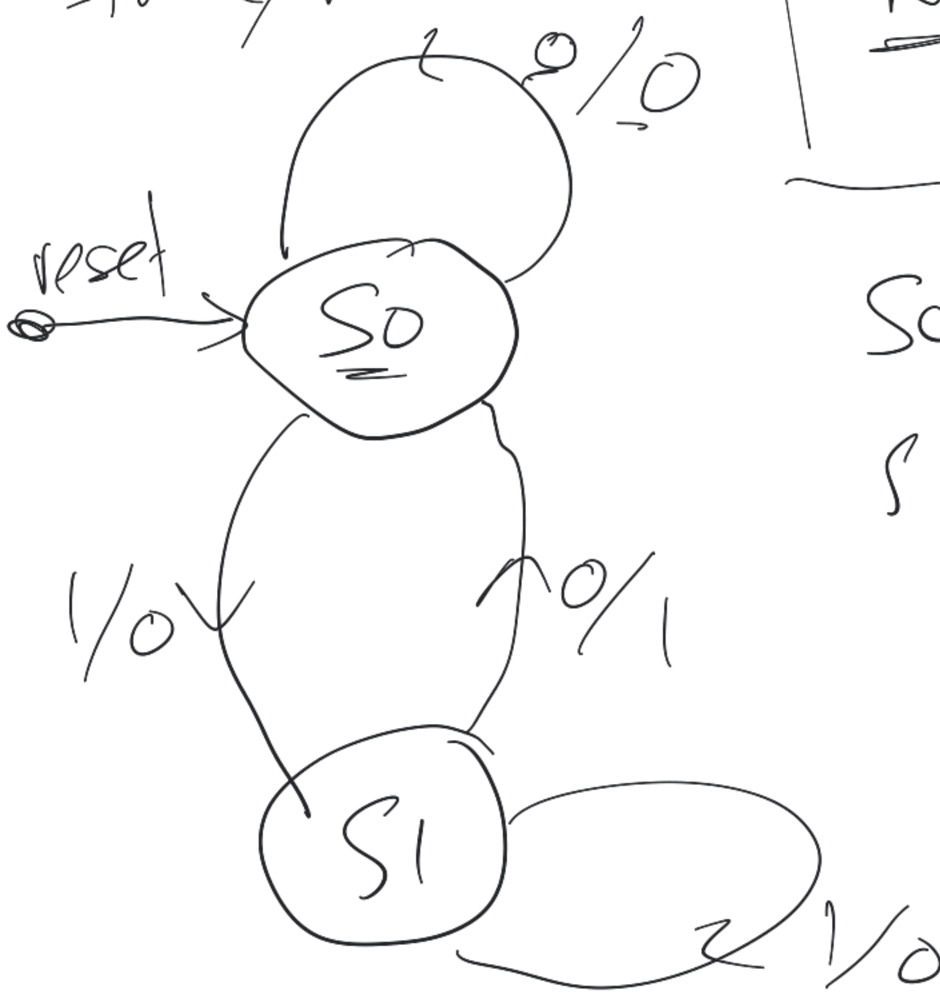
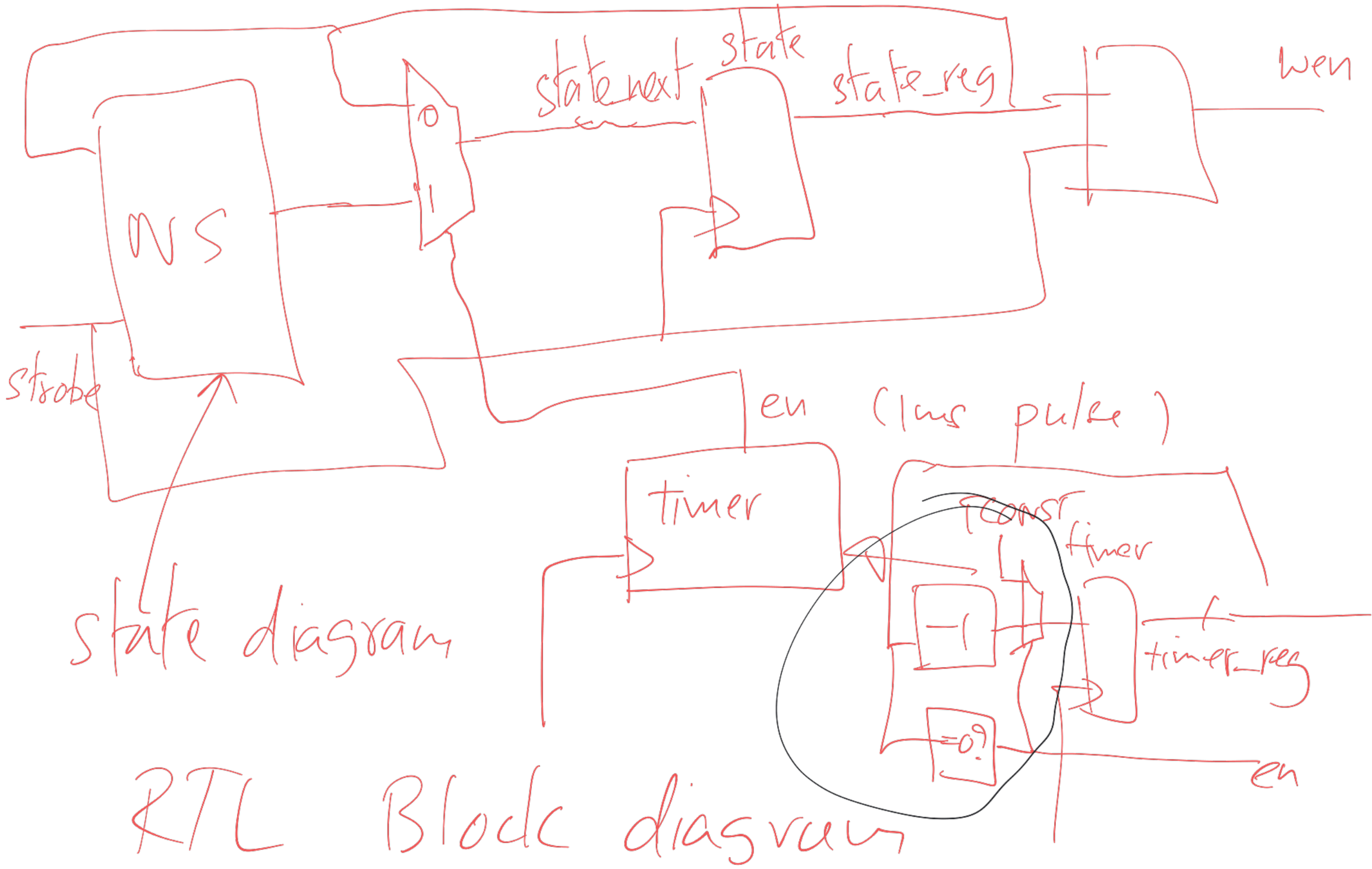

3.3V
strobe
wen
clk_25
strobe/wen


wen
1ms
Tclk


reset
S0
S1
0/0
1/0
0/1
1/0

S0 : [illegible]

S1 : [illegible]



state diagram

RTL Block diagram

```
arch
    type state_type is (S0, S1);
    signal state_reg, state_next : state_type;
    constant TCONST : unsigned(19 downto 0) :=
                        to_unsigned(200000-1, 20);
begin
    -- register
    process(clk_25, reset)
    begin
        if reset = '1' then
            state_reg <= S0;
            timer_reg <= TCONST;
        elsif rising(clk_25) then
            state_reg <= state_next;
            timer_reg <= timer_next;
        end if;
    end process;
```

signal timer_reg, timer_next:
unsigned(19 downto 0);

signal en : std_logic;

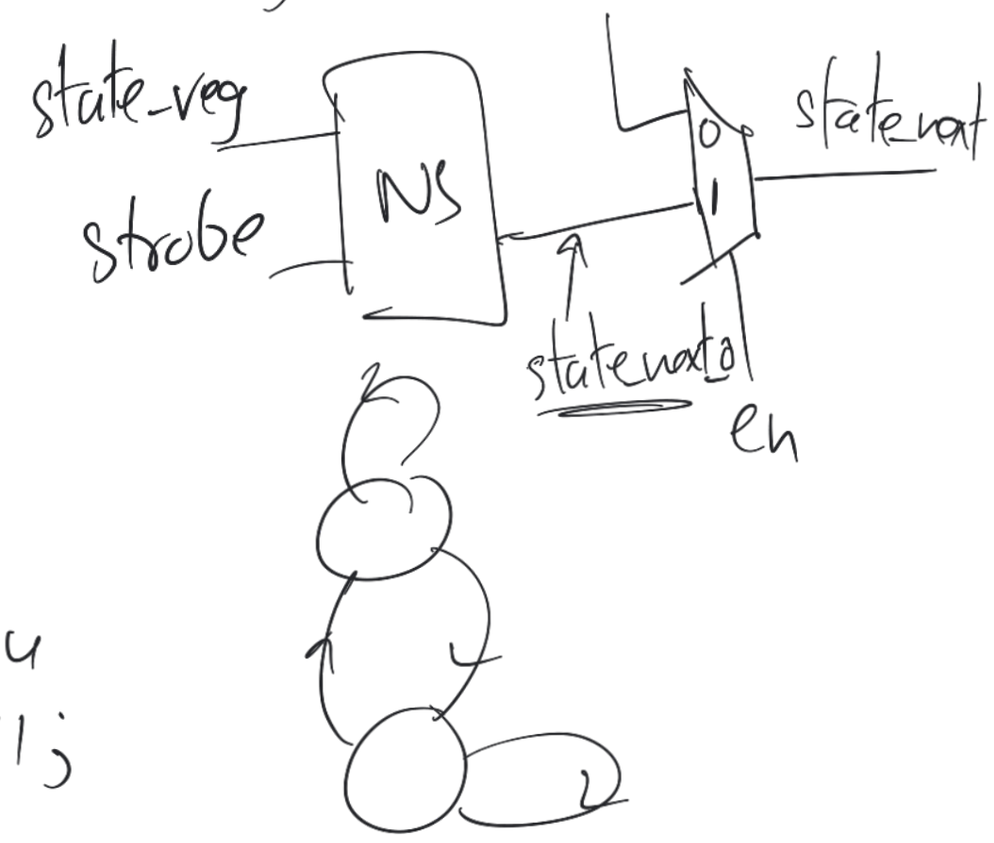
for → 0 ≠ '0'

↑ bit

timer_next ⇐ TCONST when en = '1' else
timer_reg - 1;

en ⇐ '1' when timer_reg = 0 else
'0';

process (state_reg, strobe)
begin
case state_reg is
when s0 ⇒
state_next_o ⇐ s0;
if strobe = '1' then
state_next_o ⇐ s1;
end if;
when s1 ⇒
state_next_o ⇐ s1;
if strobe = '0' then
state_next_o ⇐ s0
end if;
end case;
end process;



wen ⇐ '1' when state_reg = s1
and strobe = '0' and
en = '1' else
'0';