



Progetto 10

Semaforo

Informatica Industriale

A.A 2018/2019

Pietro Colombo

793679

Carlo Radice

807159

Outline



- Introduzione al progetto
- Specifiche implementative
- Risultati delle simulazioni
- Casi particolari

Outline



- Introduzione al progetto
- Specifiche implementative
- Risultati delle simulazioni
- Casi particolari

Introduzione al progetto

Condizioni operative del semaforo:

- NOMINAL: ROSSO acceso per 3 secondi, il VERDE per 5 secondi e il GIALLO sovrapposto al verde per 2 secondi;
- STANDBY: ROSSO e VERDE spenti e il GIALLO lampeggia, 1 secondo (acceso), 2 secondi (spento)
- MAINTENANCE: ROSSO, GIALLO e VERDE lampeggiano a intervalli regolari di 0.5 secondi. È possibile avere la durata del ROSSO pari a 6 secondi e del GIALLO pari a metà, una o due volte la durata del rosso.

All'avvio del semaforo va automaticamente in posizione MAINTENANCE.

Il semaforo è dotato dei segnali ENABLE e RESET

- ENABLE: è attivo alto, con ENABLE a '0' il semaforo è insensibile a qualsiasi controllo;
- RESET: è attivo basso, quando RESET transisce da '1' a '0' il semaforo ritorna nello stato di MAINTENANCE.

Outline



- Introduzione al progetto
- Specifiche implementative
- Risultati delle simulazioni
- Casi particolari

Specifiche implementative



- Condizione operative del semaforo rappresentate dal segnale **mode**:
 - '00' = MAINTENANCE
 - '01' = STANDBY
 - '10' = NOMINAL
- Condizioni operative modalità MAINTENANCE rappresentate dal segnale **mod_Maintenance**:
 - '00' = i segnali ROSSO, GIALLO, VERDE lampeggiano a intervalli regolari di 0.5 secondi
 - '01' = ROSSO acceso per 6 secondi, GIALLO acceso per 3 secondi
 - '10' = ROSSO acceso per 6 secondi, GIALLO acceso per 6 secondi
 - '11' = ROSSO acceso per 6 secondi, GIALLO acceso per 12 secondi

ClockFrequencyHz 100 Hz

Specifiche implementative: counter

- ad ogni ciclo di clock aggiornare i Ticks
- aggiornare i **millisecondi** quando Ticks è uguale a $\text{clockfrequency} - 10$
- aggiornare i **secondi** quando i **millisecondi** sono uguali a 900 e azzerare i **millisecondi**
- `clk` : in std_logic;
- `nRst` : in std_logic;
- `milliseconds` : inout integer := 0;
- `seconds` : inout integer := 0;
- `nRstTimer` : in std_logic

```
if rising_edge (clk) then
    if nRstTimer = '1' then
        Ticks <= 0;
        milliseconds <= 0;
        seconds <= 0;
    end if;
    -- if the negative reset signal is active
    if nRst = '0' then
        Ticks <= 0;
        milliseconds <= 0;
        seconds <= 0;
    else
        report "reset = 1";
        -- True one every 10 milliseconds
        if Ticks = ClockFrequencyHz - 10 then
            report "tick = 0";
            Ticks <= 0;
            -- True every second
            if milliseconds = 900 then
                report "aggiorno i secondi";
                milliseconds <= 0;
                seconds <= seconds + 1;
            else
                report "aggiorno i millisecondi";
                milliseconds <= milliseconds + 100;
            end if;
        else
            Ticks <= Ticks + 10;
        end if;
    end if;
end if;
```

Specifiche implementative: semaforo nominal

- Per i primi 3 secondi rimane acceso il rosso
- Per i successivi 3 secondi accendo il verde
- Per i successivi 2 secondi accendo il verde e il giallo

- clk : in std_logic;
- mode : inout std_logic_vector (1 downto 0) := "10";
- red, green, yellow : out std_logic := '0';
- nRst : in std_logic;
- milliseconds : in integer;
- seconds : in integer;
- enable : in std_logic;
- nRstTimer : inout std_logic := '0';
- mod_Maintenance : in std_logic_vector (1 downto 0)

```
if (mode = "10") then
  --nominal
  report "nominal";
  if (seconds < 3) then
    red <= '1';
    yellow <= '0';
    green <= '0';
  else
    red <= '0';
    if (seconds < 8) then
      green <= '1';
      if (seconds >= 6) then
        yellow <= '1';
      end if;
    else
      green <= '0';
      yellow <= '0';
      nRstTimer <= '1';
      red <= '1';
    end if;
  end if;
end if;
```


Specifiche implementative: semaforo standby



- Per il primo secondo accendo il giallo
- Per i successivi 3 secondi tengo tutto spento

```
if (mode = "01") then
  --standby
  report "standby";
  if (seconds < 1) then
    yellow <= '1';
    red <= '0';
    green <= '0';
  else
    if (seconds < 3) then
      yellow <= '0';
    else
      nRstTimer <= '1';
      yellow <= '1';
    end if;
  end if;
end if;
```

Specifiche implementative: semaforo maintenance

- '00' = i segnali ROSSO, GIALLO, VERDE lampeggiano a intervalli regolari di 0.5 secondi
- '01' = ROSSO acceso per 6 secondi, GIALLO acceso per 3 secondi
- '10' = ROSSO acceso per 6 secondi, GIALLO acceso per 6 secondi
- '11' = ROSSO acceso per 6 secondi, GIALLO acceso per 12 secondi

```
if (mode = "00") then
  --maintenance
  report "maintenace";
  if (mod_Maintenance = "00") then
    if (seconds = 0 and milliseconds < 500) then
      red <= '1';
      yellow <= '0';
      green <= '0';
    else
      if (seconds < 1) then
        red <= '0';
        yellow <= '1';
      else
        if (milliseconds < 500 and seconds >= 1) then
          yellow <= '0';
          green <= '1';
        else
          nRstTimer <= '1';
          green <= '0';
          red <= '1';
        end if;
      end if;
    end if;
  end if;
else
  if (seconds < 6) then
    red <= '1';
    yellow <= '0';
    green <= '0';
  else
    if ((seconds < 9 and mod_Maintenance = "01") or (seconds < 12 and mod_Maintenance = "10")
      or (seconds < 18 and mod_Maintenance = "11")) then
      -- in base alla modalita la durata del giallo cambia
      red <= '0';
      yellow <= '1';
    else
      nRstTimer <= '1';
      yellow <= '0';
      red <= '1';
    end if;
  end if;
end if;
else
```

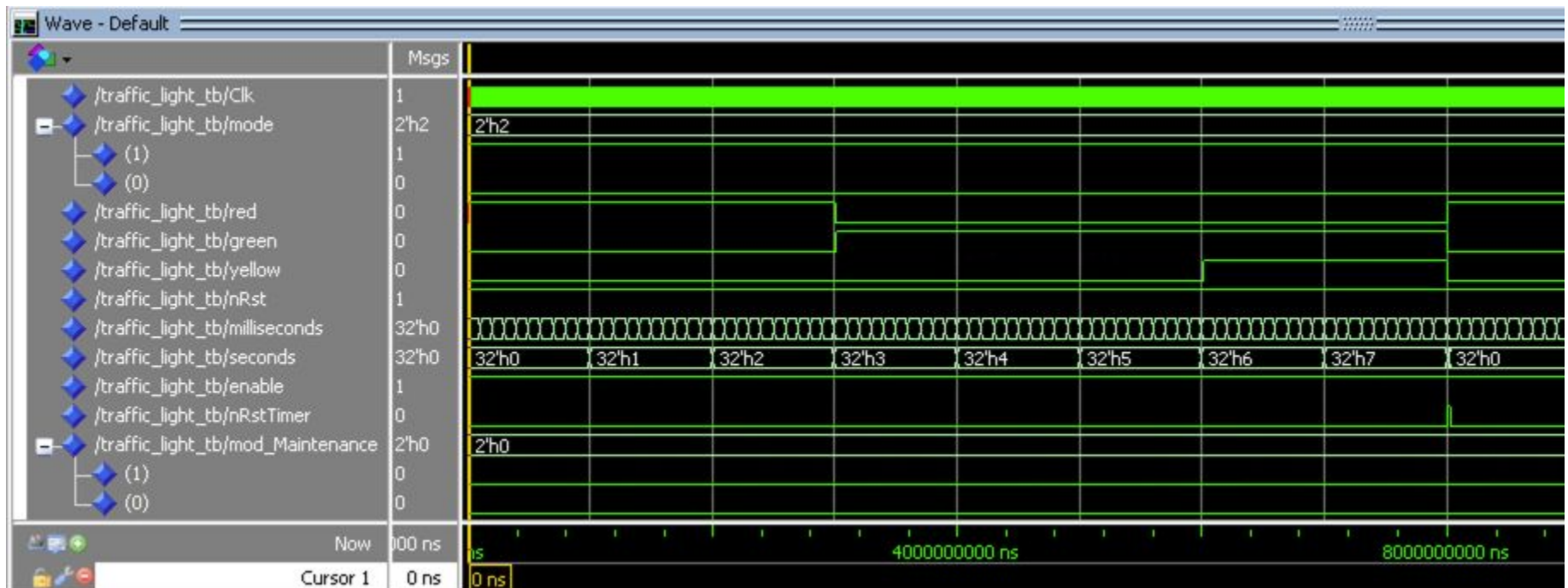
Outline



- Introduzione al progetto
- Specifiche implementative
- Risultati delle simulazioni
- Casi particolari

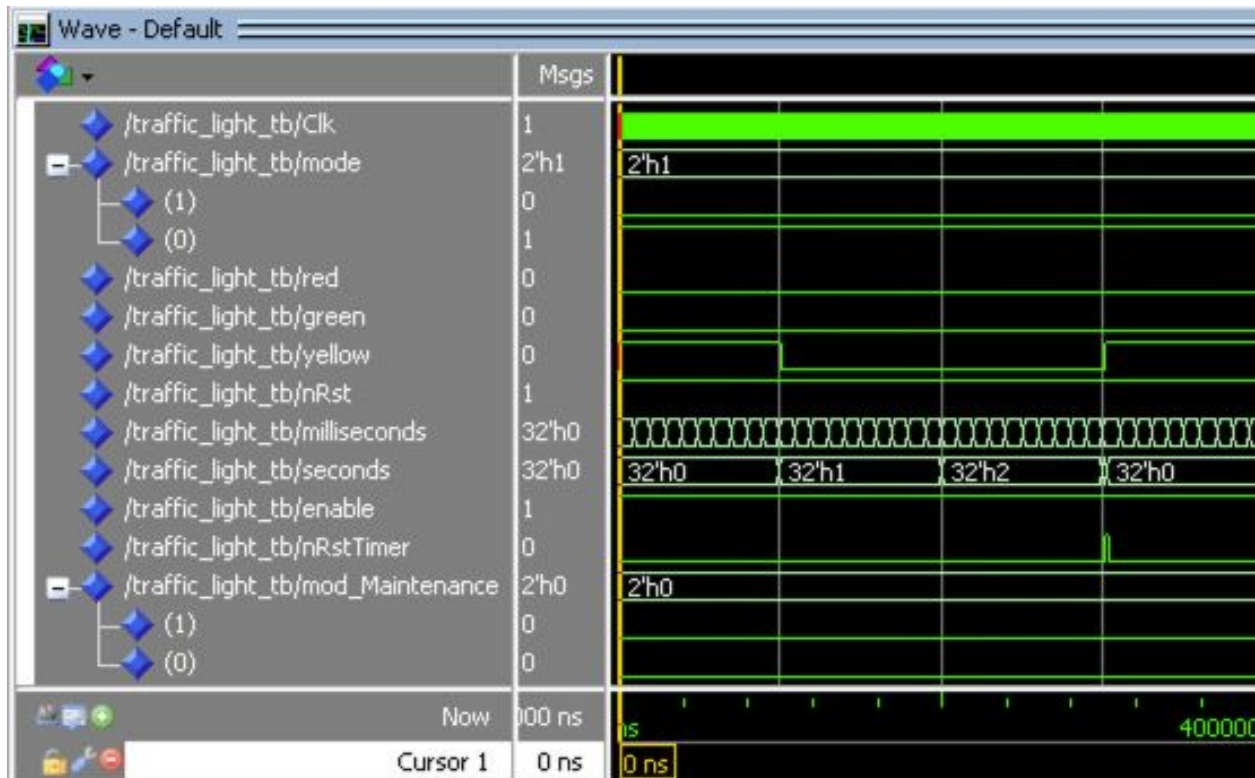
Risultati e Simulazioni

- '10' = NOMINAL: ROSSO on 3 sec, VERDE on 5 sec, GIALLO sovrapposto al verde per 2 sec.



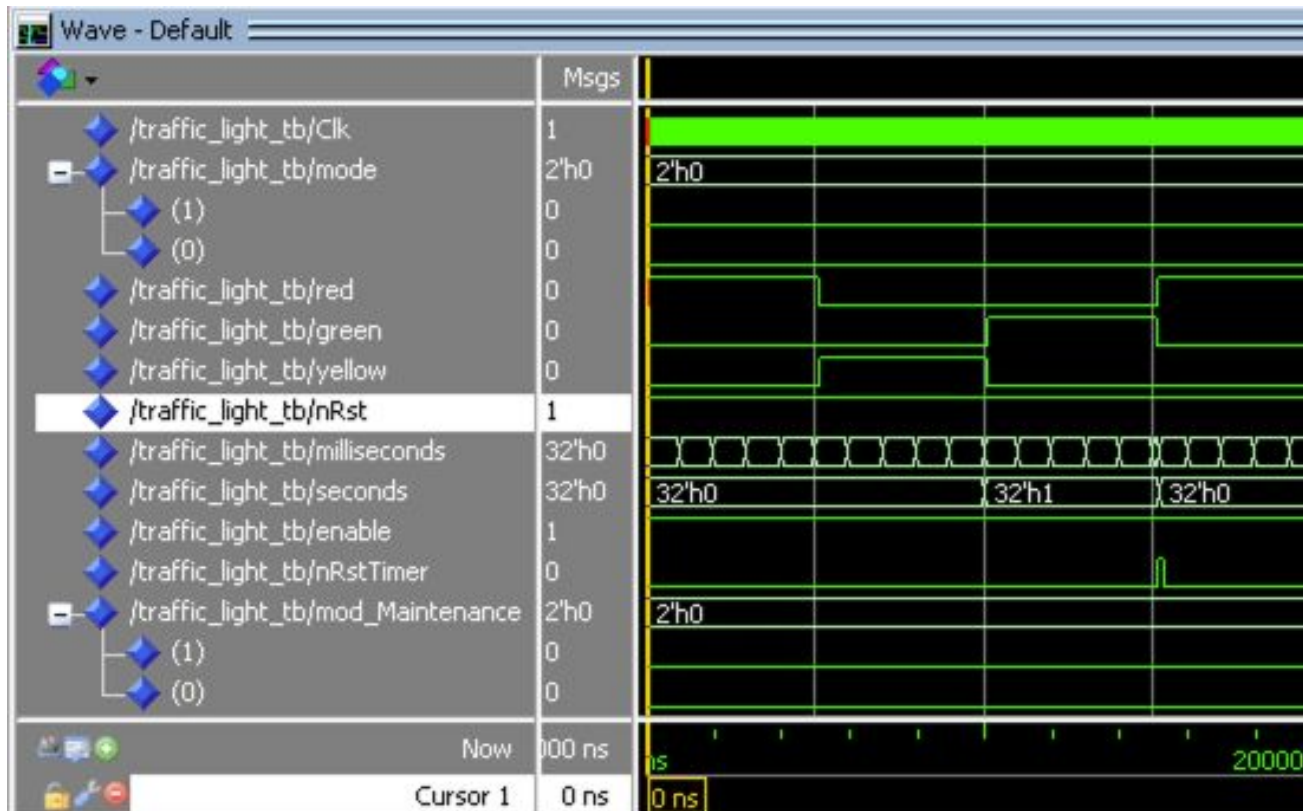
Risultati e Simulazioni

- '01' = STANDBY: GIALLO on 1 sec, GIALLO off 2 sec.



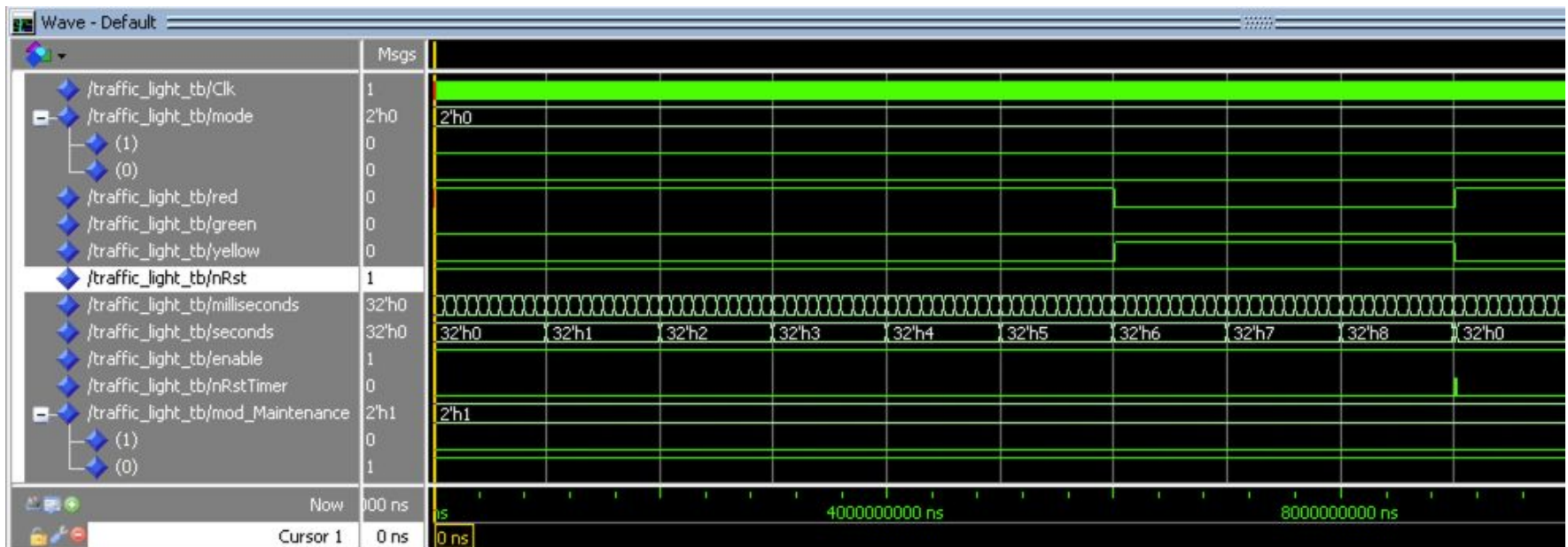
Risultati e Simulazioni

- '00' = MAINTENANCE:
 - '00' = i segnali ROSSO, GIALLO, VERDE lampeggiano a intervalli regolari di 0.5 secondi



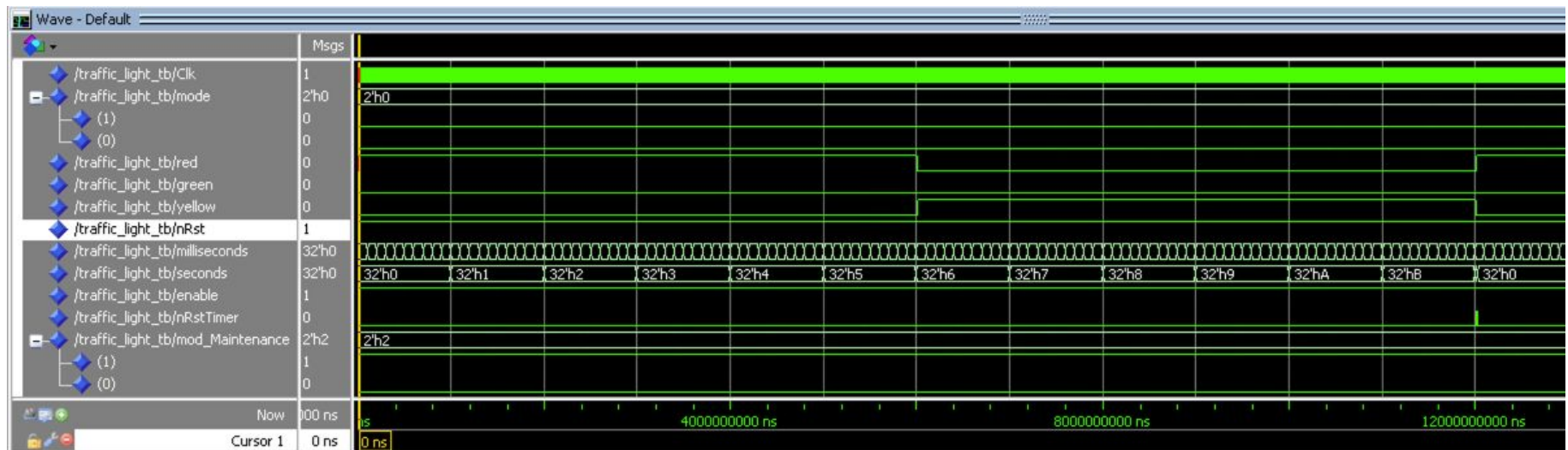
Risultati e Simulazioni

- '00' = MAINTENANCE:
 - '01' = ROSSO acceso per 6 secondi, GIALLO acceso per 3 secondi



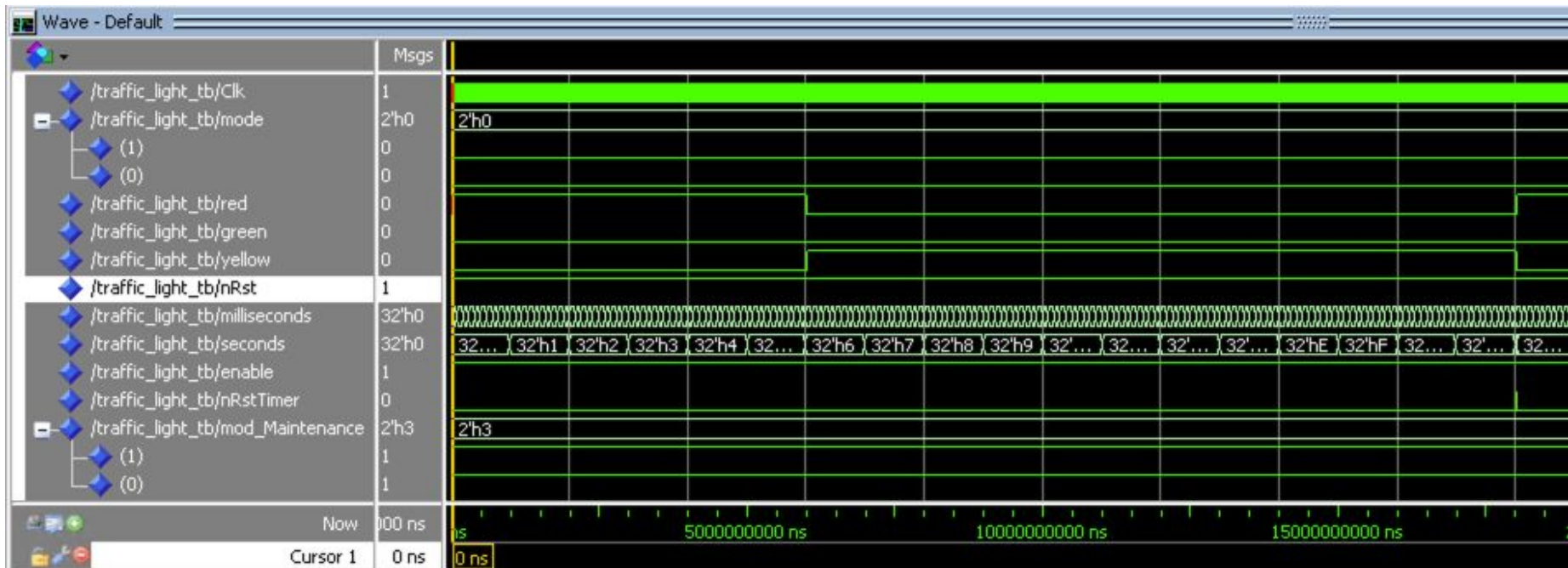
Risultati e Simulazioni

- '00' = MAINTENANCE:
 - '10' = ROSSO acceso per 6 secondi, GIALLO acceso per 6 secondi



Risultati e Simulazioni

- '00' = MAINTENANCE:
 - '11' = ROSSO acceso per 6 secondi, GIALLO acceso per 12 secondi



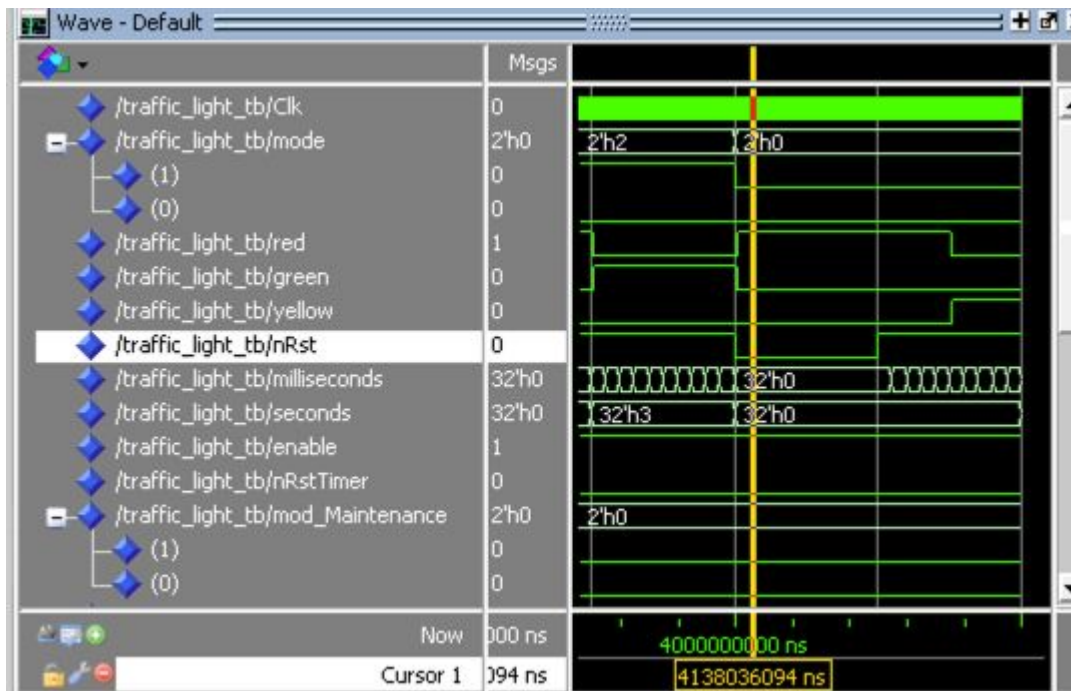
Outline



- Introduzione al progetto
- Specifiche implementative
- Risultati delle simulazioni
- Casi particolari

Casi Particolari

- Reset attivo basso, passo alla modalità Maintenance (lampeggiano ad intervalli regolari di 0.5 secondi)



Conclusioni



Il lavoro effettuato consiste nell'implementazione di un semaforo

- Il sistema si comporta come definito da progetto secondo le varie modalità.

Dopo aver effettuato le simulazioni

- i risultati sono coerenti con quanto richiesto dal progetto