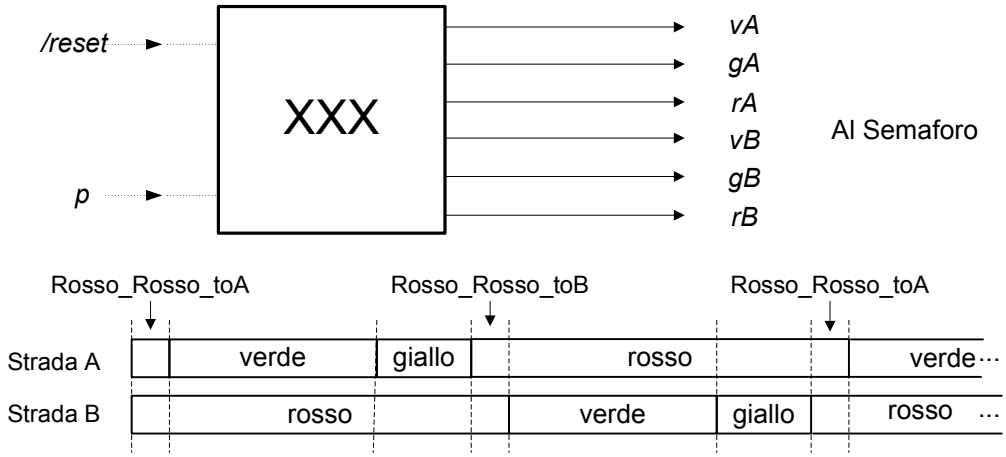
Reti Logiche 08/06/2012
Nome e Cognome: _____ Matricola: _____

Esercizio 1
Descrivere e sintetizzare un *verificatore di handshake*. Tale rete è una rete sequenziale asincrona con due ingressi `dav_` e `rfd` e un'uscita `error`, che analizza se gli handshake tra due altre reti avvengono correttamente. Si comporta come segue:

- Fintanto che gli handshake si stanno svolgendo in maniera corretta, l'uscita vale zero.
- Non appena le variabili di ingresso hanno una transizione non consentita dalle regole dell'handshake, la rete porta l'uscita ad 1 e *resta bloccata in eterno*.

Si assuma che al reset gli ingressi valgano 11.

Esercizio 2
Descrivere l'Unità XXX che comanda un semaforo all'incrocio di due strade A e B in accordo alle specifiche riportate nella figura (*vA* a 1 accende il verde nel percorso relativo alla strada A, *gA* a 1 accende il giallo nel percorso relativo alla strada A, *rA* a 1 accende il rosso nel percorso relativo alla strada A, …).



Nota:
1) Far durare il verde per 20 cicli di clock, il giallo per 3 cicli di clock ed il rosso-rosso per 1 ciclo di clock
2) Inizializzare l'Unità XXX nello stato interno *S*0 indicante *Rosso_Rosso_toA*:

ovvero

```
module XXX(vA,gA,rA, vB,gB,rB, p,reset_);
 input       p,reset_;
 output      vA,gA,rA, vB,gB,rB;
 reg         VA,GA,RA, VB,GB,RB; assign vA=VA,gA=GA,rA=RA, vB=VB,gB=GB,rB=RB;
 reg [...:0] COUNT;
 reg [2:0]   STAR; parameter  S0=0, S1=1, ... ;
 always @(posedge p or negedge reset_)
  if (reset_==0) begin VA=0; VB=0; GA=0; GB=0; RA=1; RB=1; STAR=S0; end else #3
   casex(STAR)
    S0:  ...
    ...
  endcase
endmodule
```

<u>Fondamentale</u>: Simulare l'evoluzione dell'Unità da voi descritta ammettendo, per brevità, che il verde duri 4 cicli di clock ed il giallo 2 cicli di clock.

Reti Logiche 08/06/2012

Nome e Cognome: _____ Matricola: _____