Grammatica dichiarazione:

$S \longrightarrow$ "module" N "(" "/n" E

$E \longrightarrow$ "clk" "/n" E_1 | E_1

$E_1 \longrightarrow$ "input" I "/n" E_1 | "input" T_1 | "/n" E_1

$T_1 \longrightarrow I$ "," T_1 | I "/n" O

$O \longrightarrow$ "output" I "/n" O | output" I "/n" ");" | "ouput" T_O | "/n" O

$T_O \longrightarrow I$ "," T_O | I "/n" O

(*Dove I è il generico identidicatore che non può essere flip flop)

Grammatica instance:

$S \longrightarrow id$ "instance" "(" T ")"

$T \longrightarrow$ "." id "=" input T_1 T_O

$T_1 \longrightarrow$ "," "." id "=" input T_1 | ε

$T_O \longrightarrow$ "," "." id "=" C T_O | ε

$C \longrightarrow input$ | $flip\_flop$

SDT espressioni booleane:

$I \longrightarrow H$ [I'.prec=H.val] I' [I.val=S'.val]

$I' \longrightarrow NOR\ H$ [I'1.prec= I'.prec nor H.val] I'1[I'.val=I'1.val]

$I' \longrightarrow \varepsilon$ [I'.val = I'.prec]

$H \longrightarrow E$ [I'.prec=H.val] H' [I.val=S'.val]

$H' \longrightarrow XOR\ E$ [H'1.prec= H'.prec xor E.val] H'1[H'.val=H'1.val]

$H' \longrightarrow \varepsilon$ [H'.val = H'.prec]

$E \longrightarrow G$ [E'.prec=G.val] E' [E.val=S'.val]

$E' \longrightarrow OR\ G$ [E'1.prec= E'.prec or G.val] E'1[E'.val=E'1.val]

$E' \longrightarrow \varepsilon$ [E'.val = E'.prec]

$G \longrightarrow T$ [G'.prec=T.val] G' [G.val=G'.val]

$G' \longrightarrow NAND\ T$ [G'1.prec= G'.prec nand T.val] G'1[G'.val=G'1.val]

$G' \longrightarrow \varepsilon$ [G'.val = G'.prec]

$T \longrightarrow F$ [T'.prec=F.val] T' [T.val=T'.val]

$T' \longrightarrow AND\ T$ [T'1.prec= T'.prec and T.val] T'1[T'.val=T'1.val]

$T' \longrightarrow \varepsilon$ [T'.val = T'.prec]

$F \longrightarrow N\ F'$ [if(N.val=1)F.val= !F'.val else F.val= F'.val]

$F' \longrightarrow (I)$ [F'.val=I.val]

$F' \longrightarrow id$

$N \longrightarrow NOT$ [N.val=1]

$N \longrightarrow \epsilon$ [N.val=0]