Să se implementeze un modul numit **ierdna**.

   Modulul va avea următoarele intrări: **aral**, **atac**, **ide**, **icul** și ieșiri **anao**, **anaid**, **luap**.

*Atenție! Valorile intărilor vor fi setate în testbench.*

a) **aral**, **atac**, **ide** și **icul** vor fi inițial setate pe **0;**

       b) după o unitate de timp, **aral** și **atac** se vor seta pe **1;**

       c) după altă unitate de timp, **aral** și **ide** se vor seta pe **0,** respectiv **1;**

       d) după altă unitate de timp, toate intrările vor fi setate pe **1**.

        Modulul va reacționa doar când **ide** sau **icul** se vor modifica. *(Sfat: această condiție se va implementa in***always***)*

Ieșirea **anao** va fi egală cu suma dintre **icul** și **atac** dacă **aral** este **0**, iar dacă **aral** este **1**, **anao** va fi egală cu diferența dintre **ide** și **atac**.

Ieșirea **anaid** va fi **atac\* icul** dacă **icul** este **0** și **aral= 1**, iar in rest **atac\icul**.

Ieșirea **luap** va fi egală cu suma dintre **aral**, **atac**, **ide** și **icul**.

*Analizați wave-urile și spuneți ce observați. Încercați alte valori pentru datele de intrare și spuneți dacă există și cazuri care nu sunt bine definite.*

ierdna.v

module ierdna(

input aral, atac, ide, icul,

output reg anao, anaid, luap);

always @(ide or icul)begin

if(aral==0)

anao = icul + atac;

else

anao = ide - atac;

if(icul==0)

anaid = atac\*icul;

else

anaid = atac/icul;

luap=aral+atac+ide+icul;

end

endmodule

ierdna\_tb.v:

module ierdna\_tb;

reg aral\_tb, atac\_tb, ide\_tb, icul\_tb;

wire anao\_tb, anaid\_tb, luap\_tb;

initial begin $monitor("Timp=%d,ide=%b,icul=%b,anao=%b,anaid=%b,luap=%b", $time, ide\_tb, icul\_tb,anao\_tb,anaid\_tb,luap\_tb);

end

initial begin

aral\_tb=0;

atac\_tb=0;

ide\_tb=0;

icul\_tb=0;

#1 aral\_tb =0;

atac\_tb=1;

#1 aral\_tb=0;

ide\_tb=1;

#1 aral\_tb=1;

atac\_tb=1;

ide\_tb=1;

icul\_tb=1;

end

ierdna poftim(

.aral(aral\_tb),

.atac(atac\_tb),

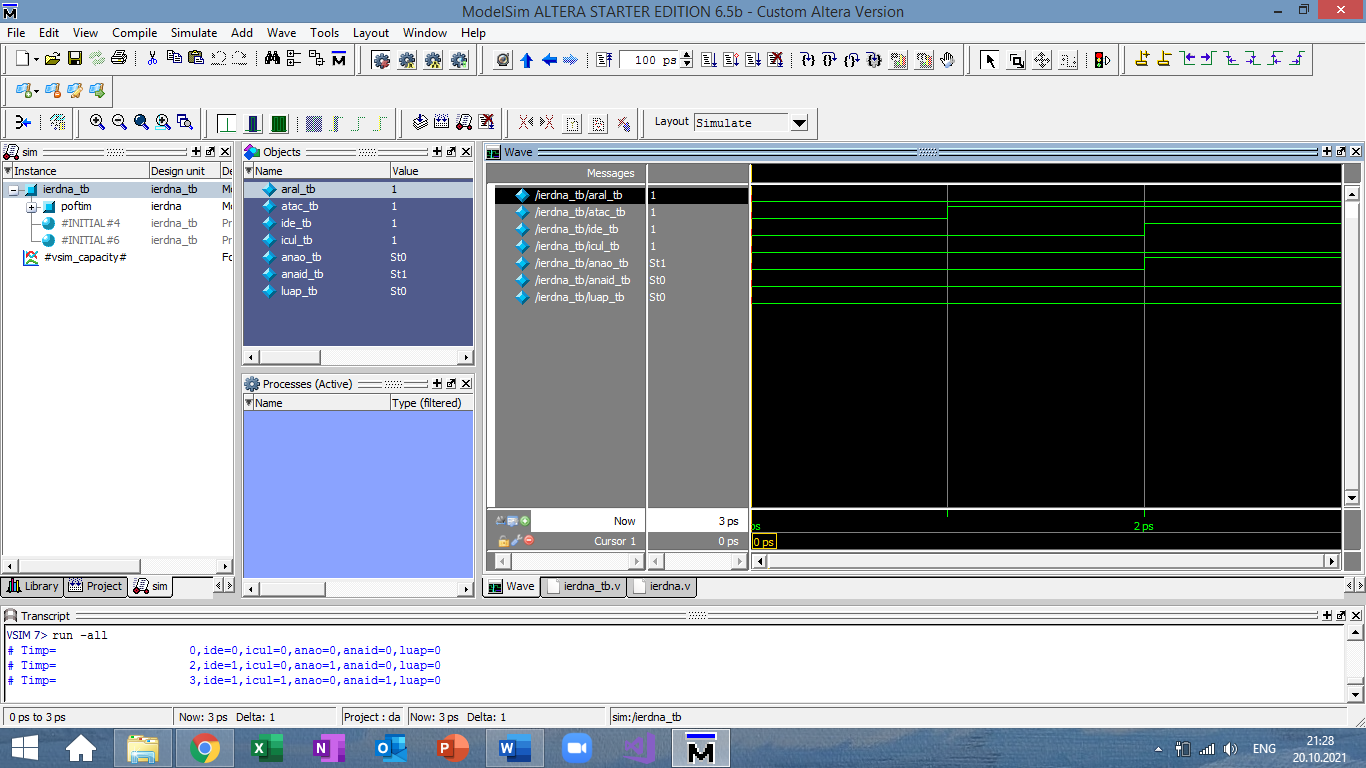
.ide(ide\_tb),

.icul(icul\_tb),

.anao(anao\_tb),

.anaid(anaid\_tb),

.luap(luap\_tb));

endmodule

După cum putem observa şi în poza cu wave-ul, iniţial toate variabilele sunt setate la 0. După o unitate de timp, doar variabila atac\_tb ia valoarea 1, deci wave-ul specific acestuia se ridică. După o altă unitate de timp, wave-ul variabilei atac\_tb rămâne tot pe valoarea 1 şi de asemenea creşte şi cel de la variabila ide\_tb şi anao\_tb.

Orice valori i-am da programului (dintre 0 şi 1), acesta rulează pozitiv. Ar fi apărut erori în momentul în care s-ar fi împărţit la 0, însă această variantă este ocolită datorită structurii:

if(icul==0)

anaid = atac\*icul;

else

anaid = atac/icul;