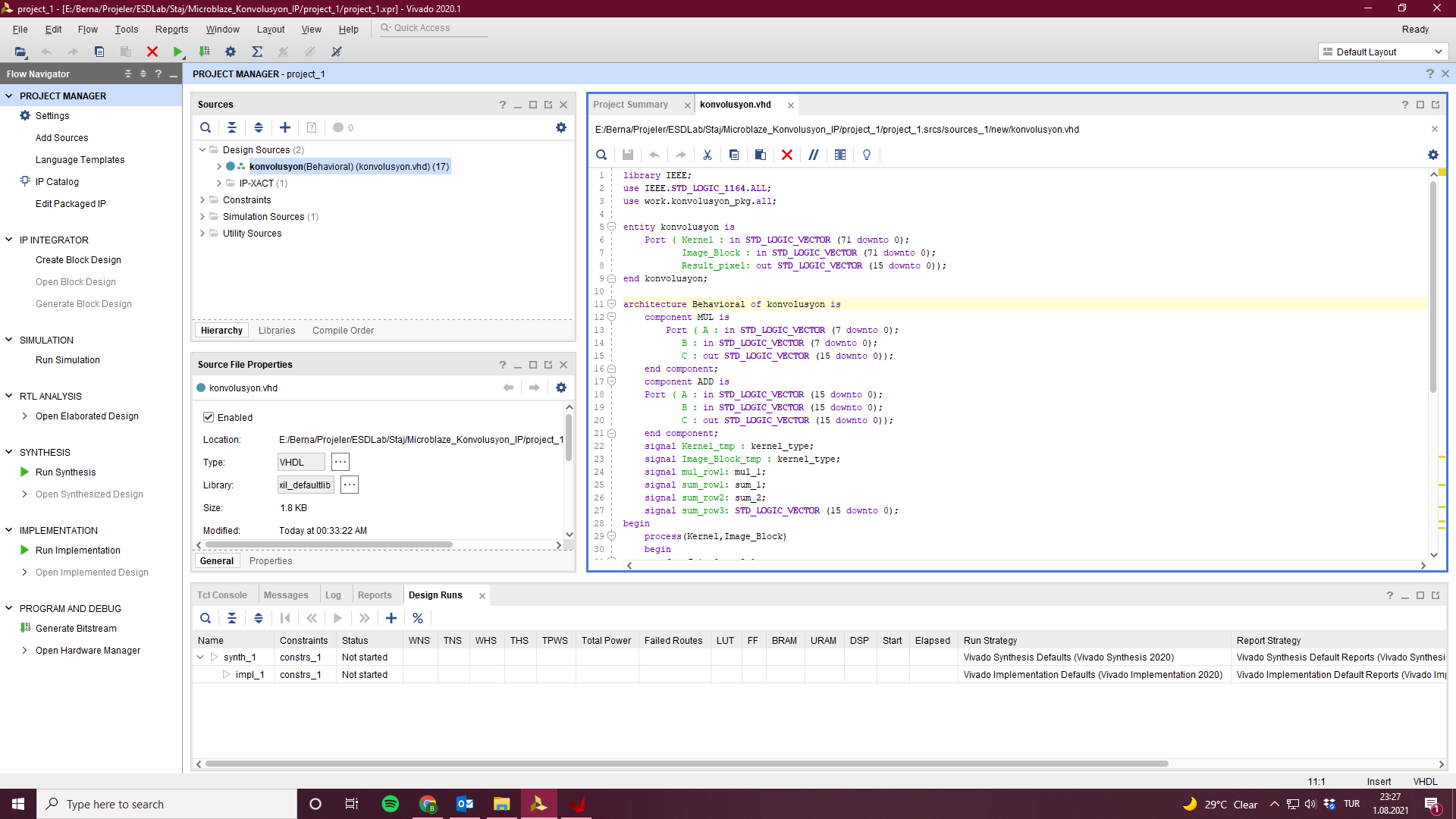
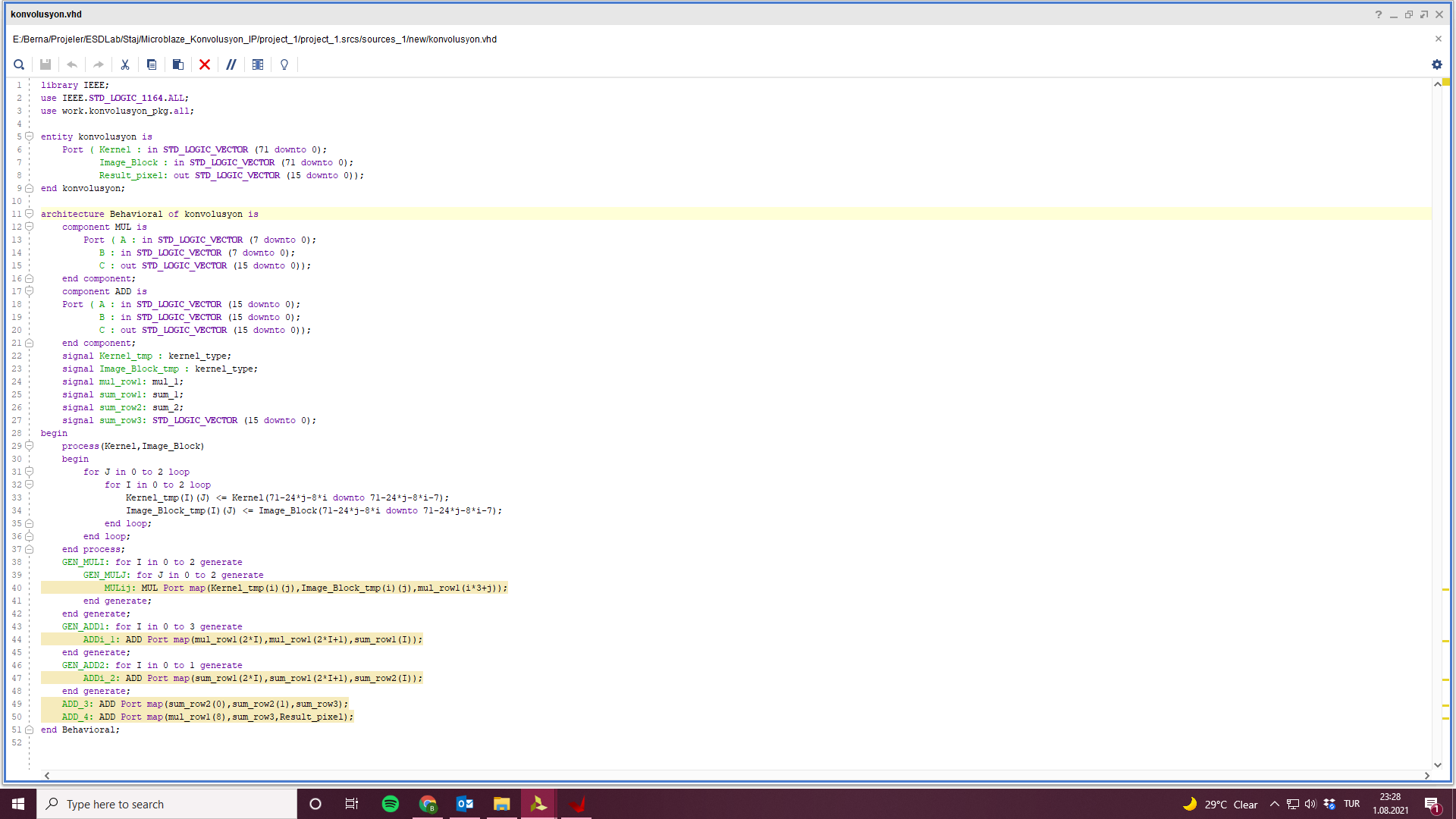
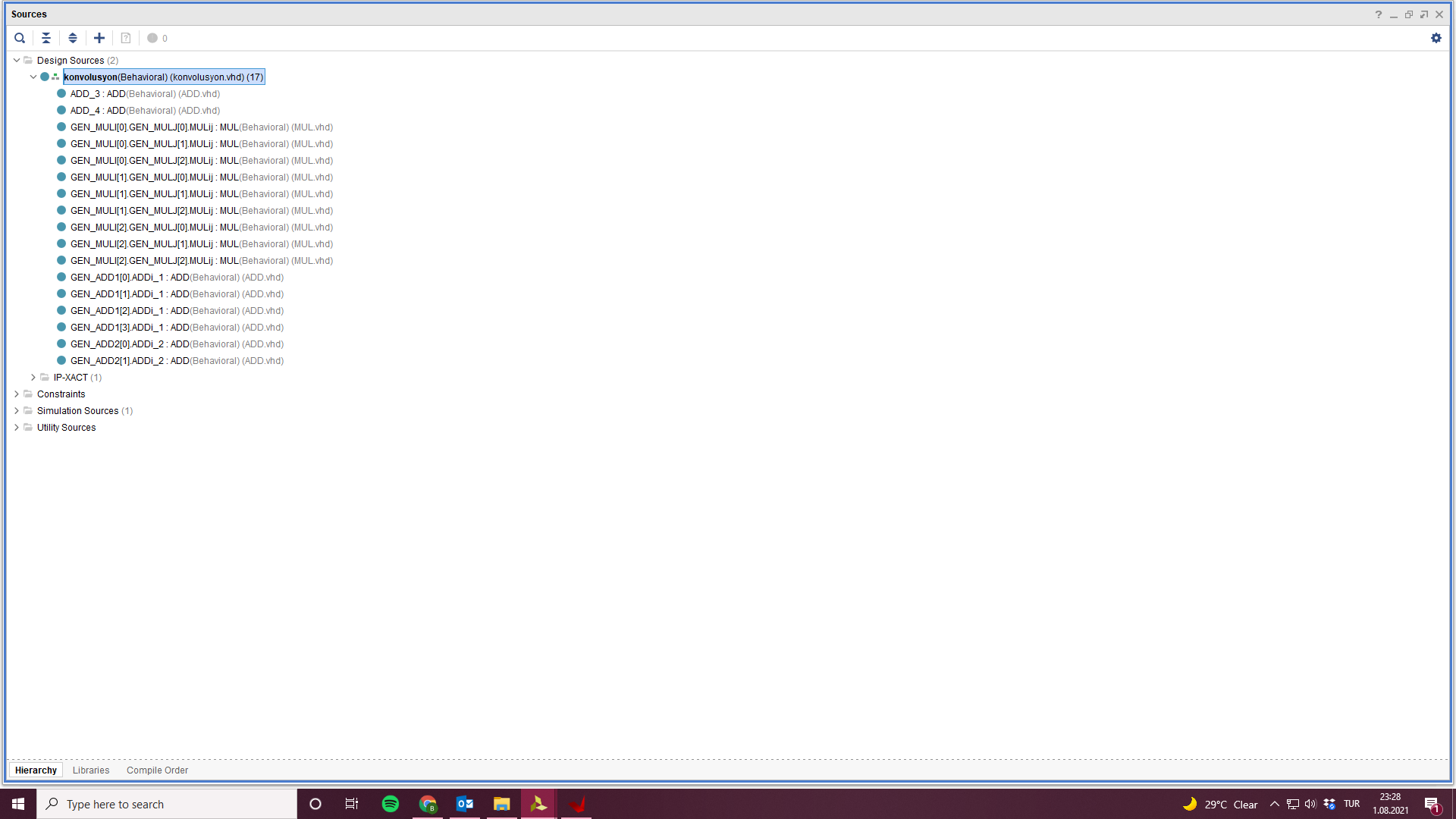
42) Şimdi konvolüsyon işlemini Custom IP de gerçekleştirmeyi deneyeceğim.

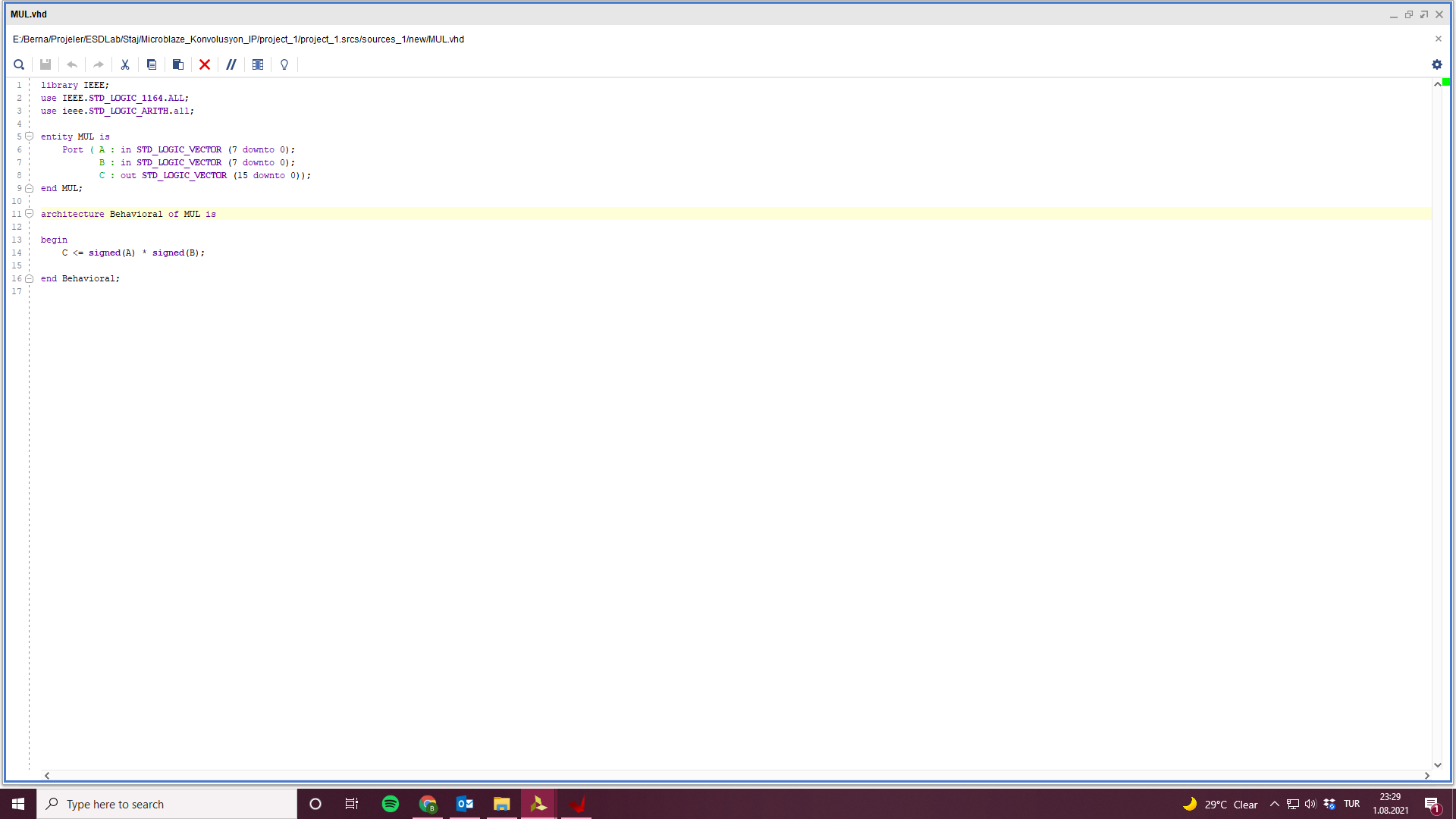
<https://www.xilinx.com/video/hardware/creating-an-axi-peripheral-in-vivado.html>

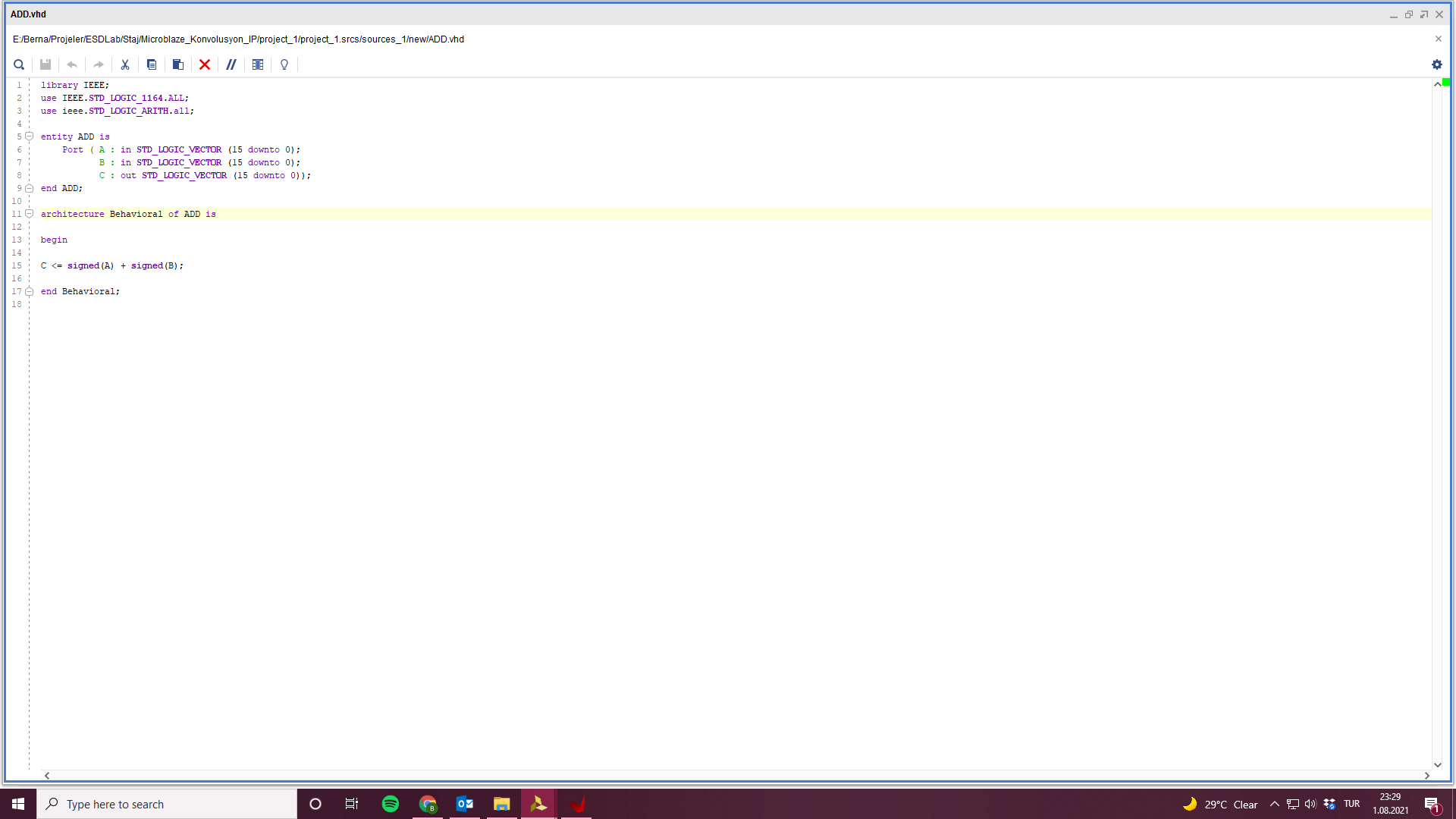
İlk olarak 3x3 konvolüsyonu gerçekleyecek bir devre tasarladım. Bunun için bir Vivado projesi oluşturdum ve VHDL kodlarını yazdım.

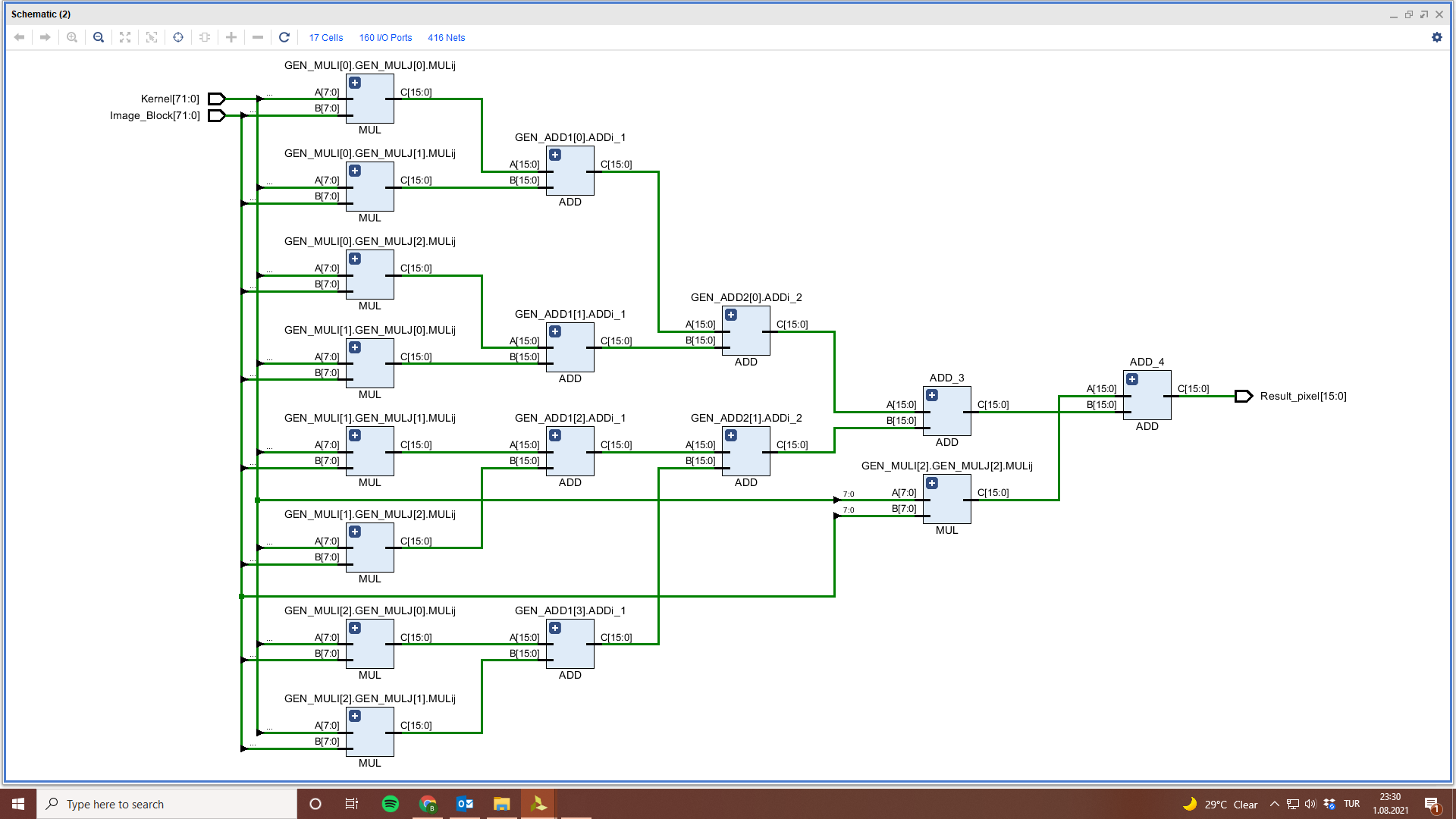




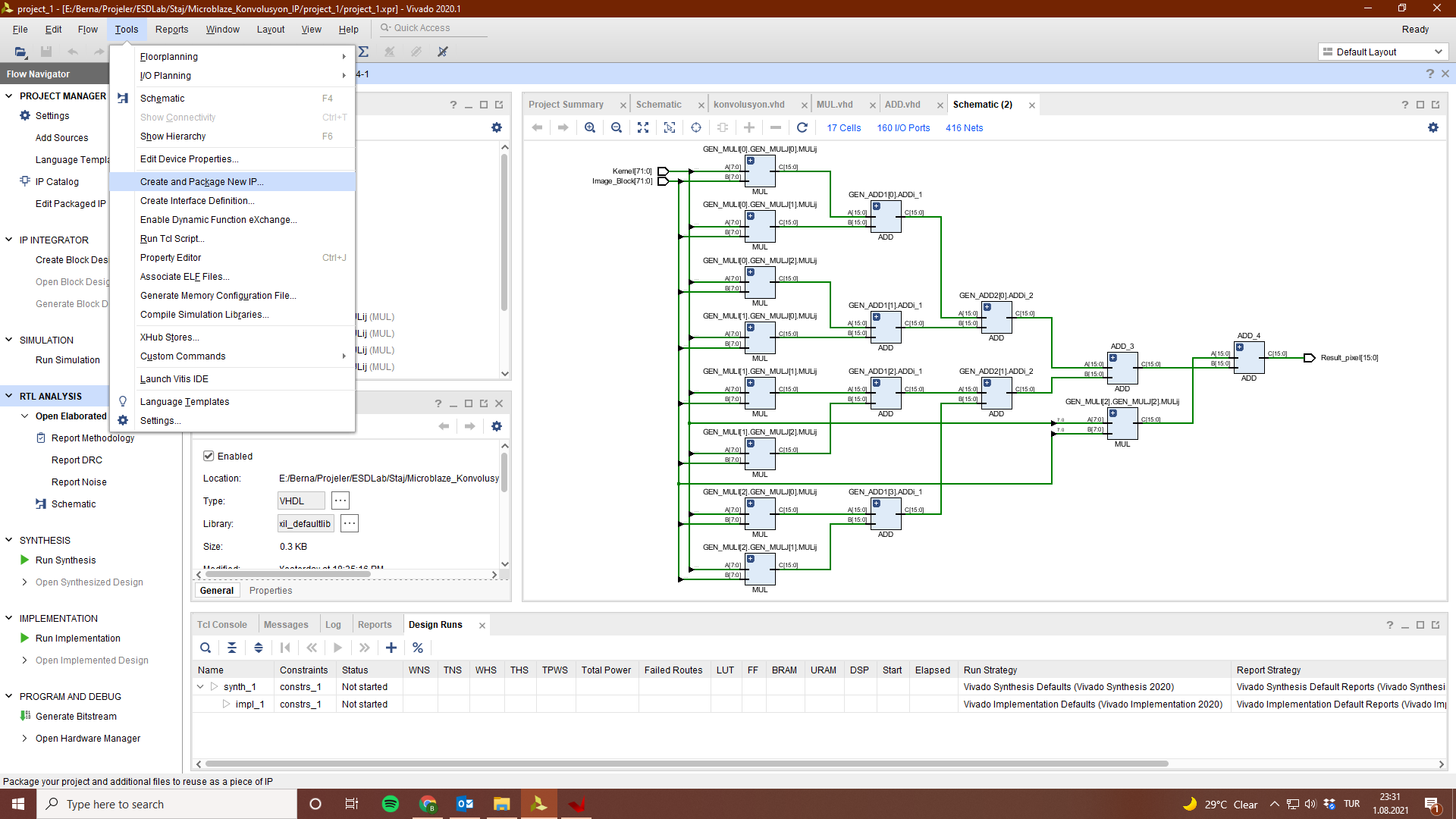


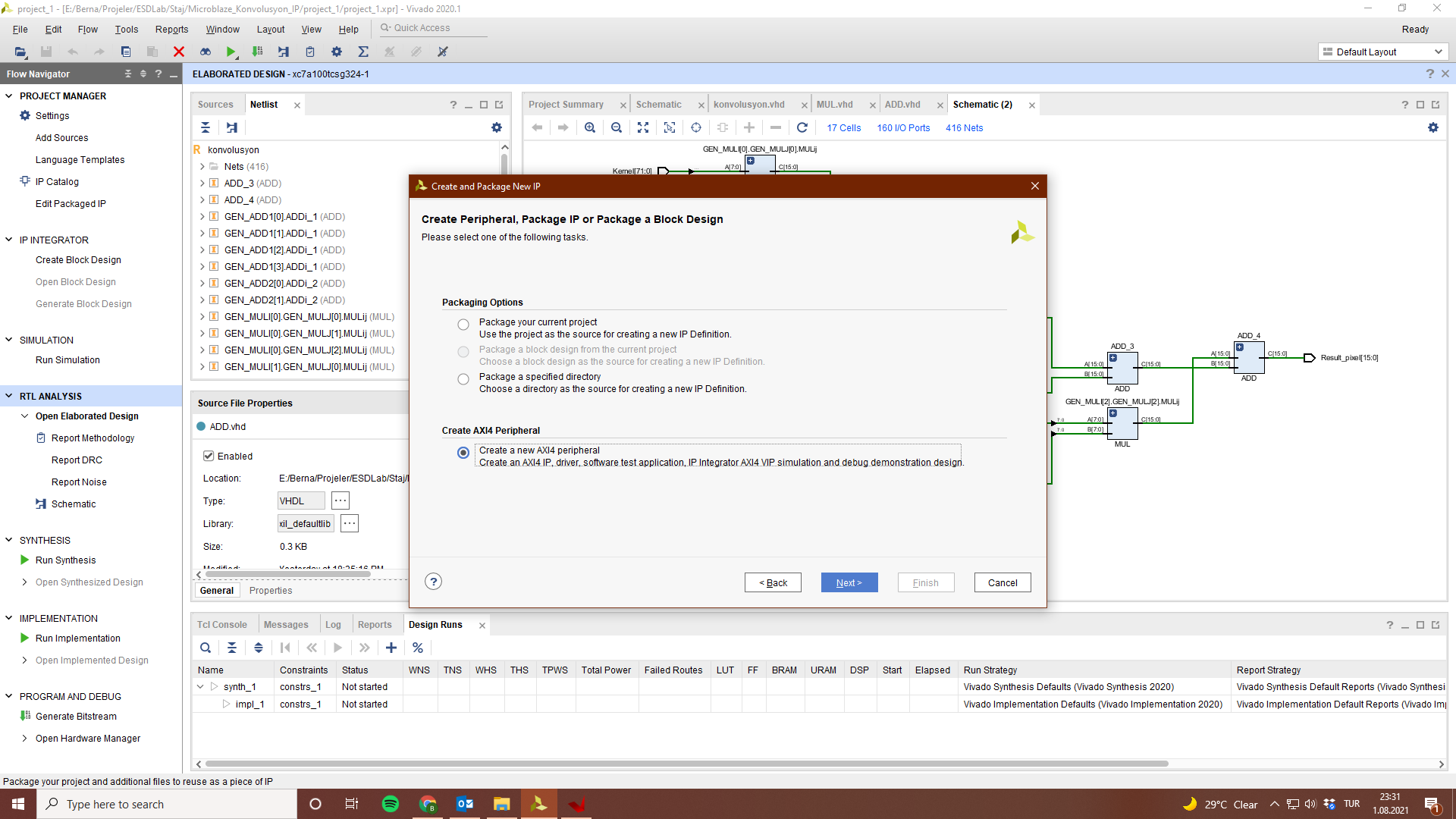


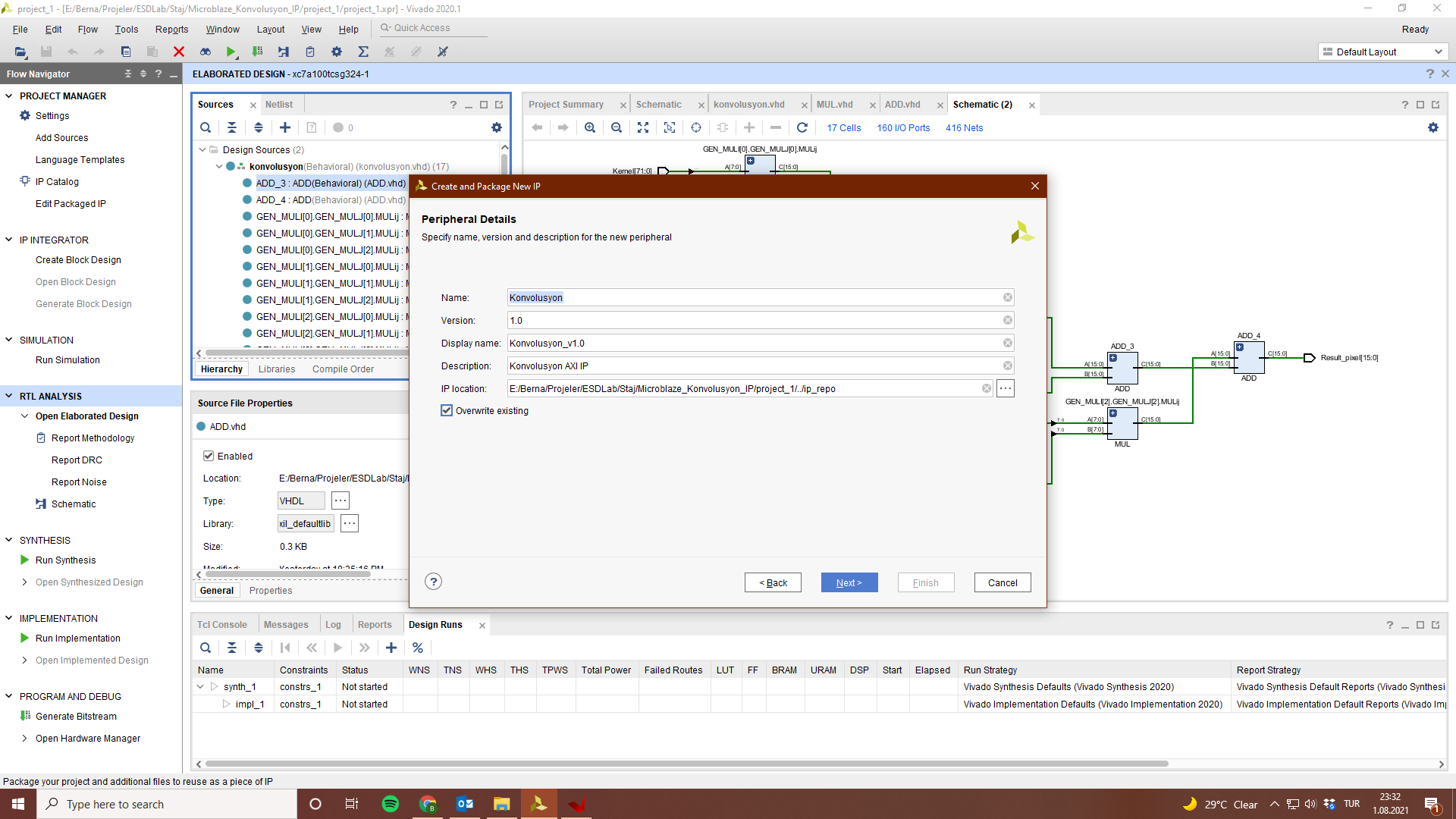


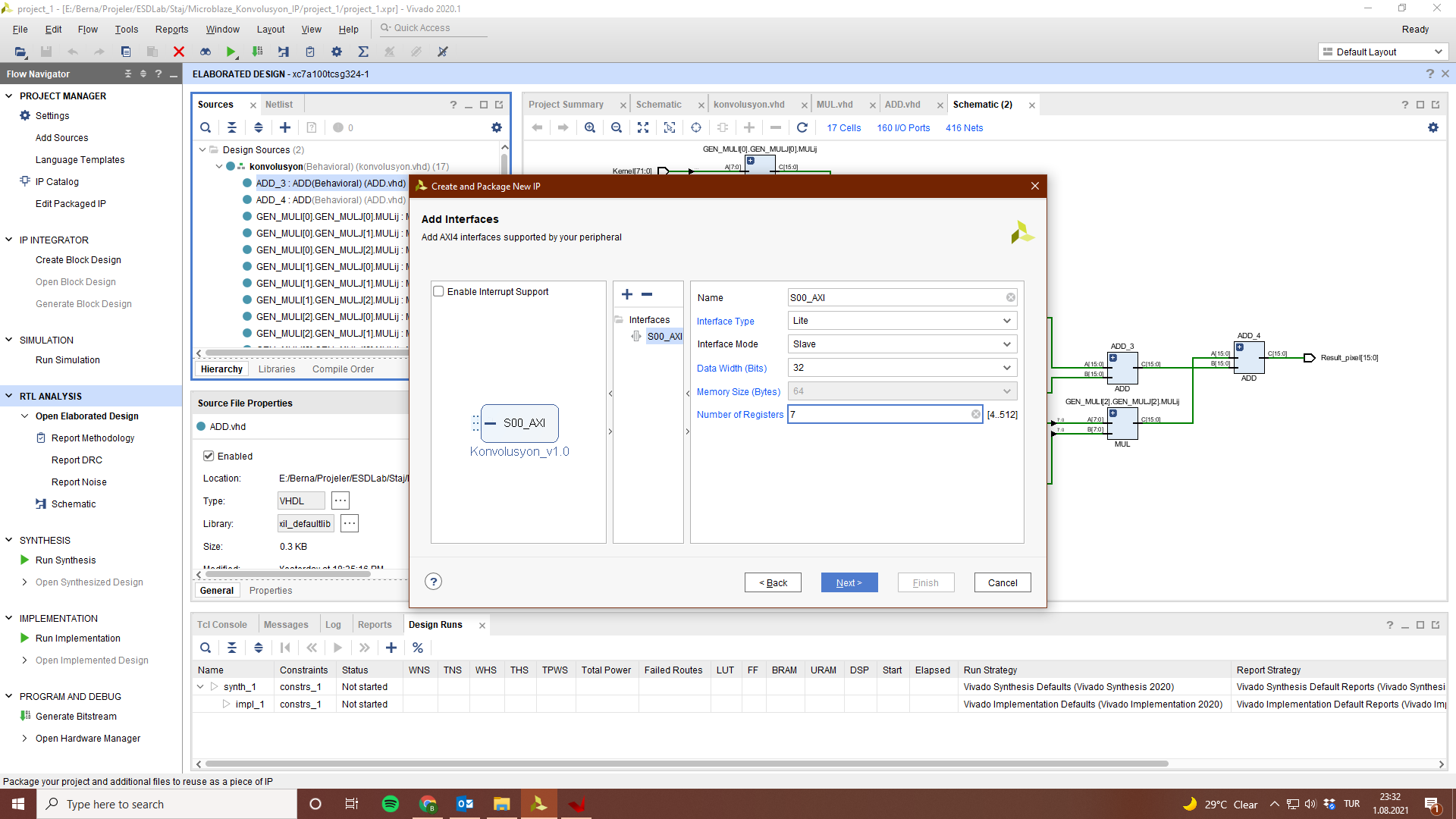


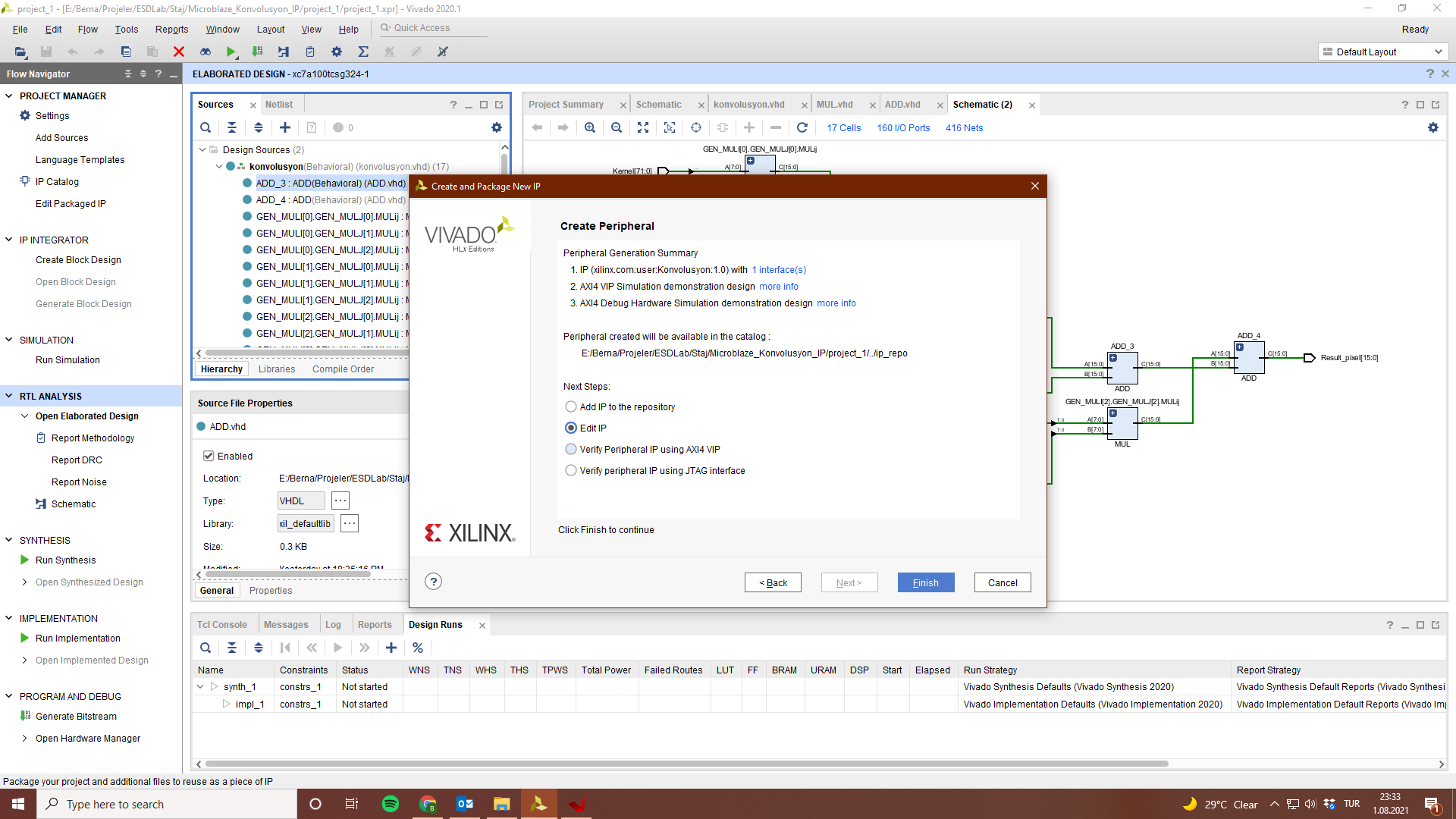
43) Create and Package New IP

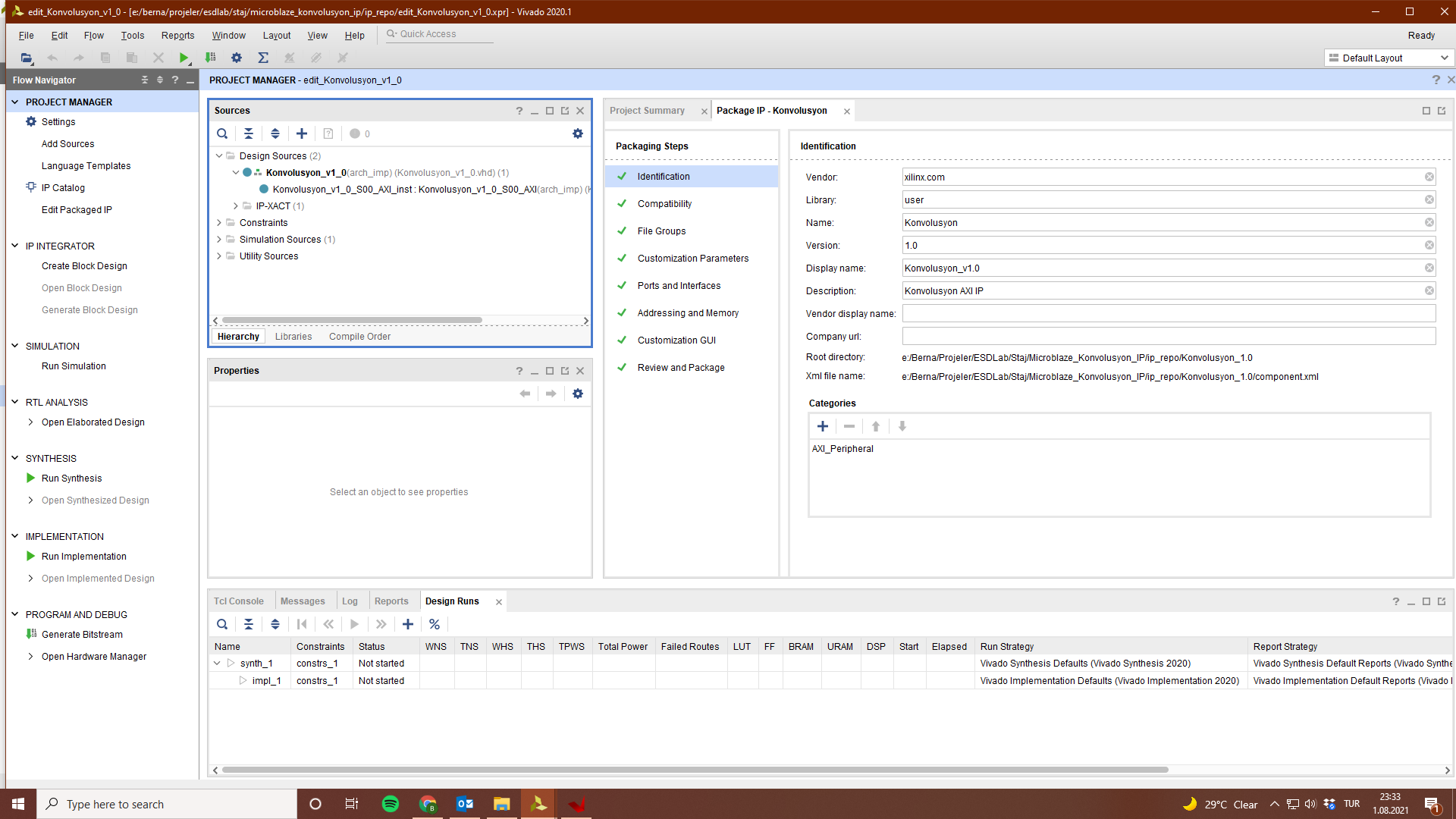


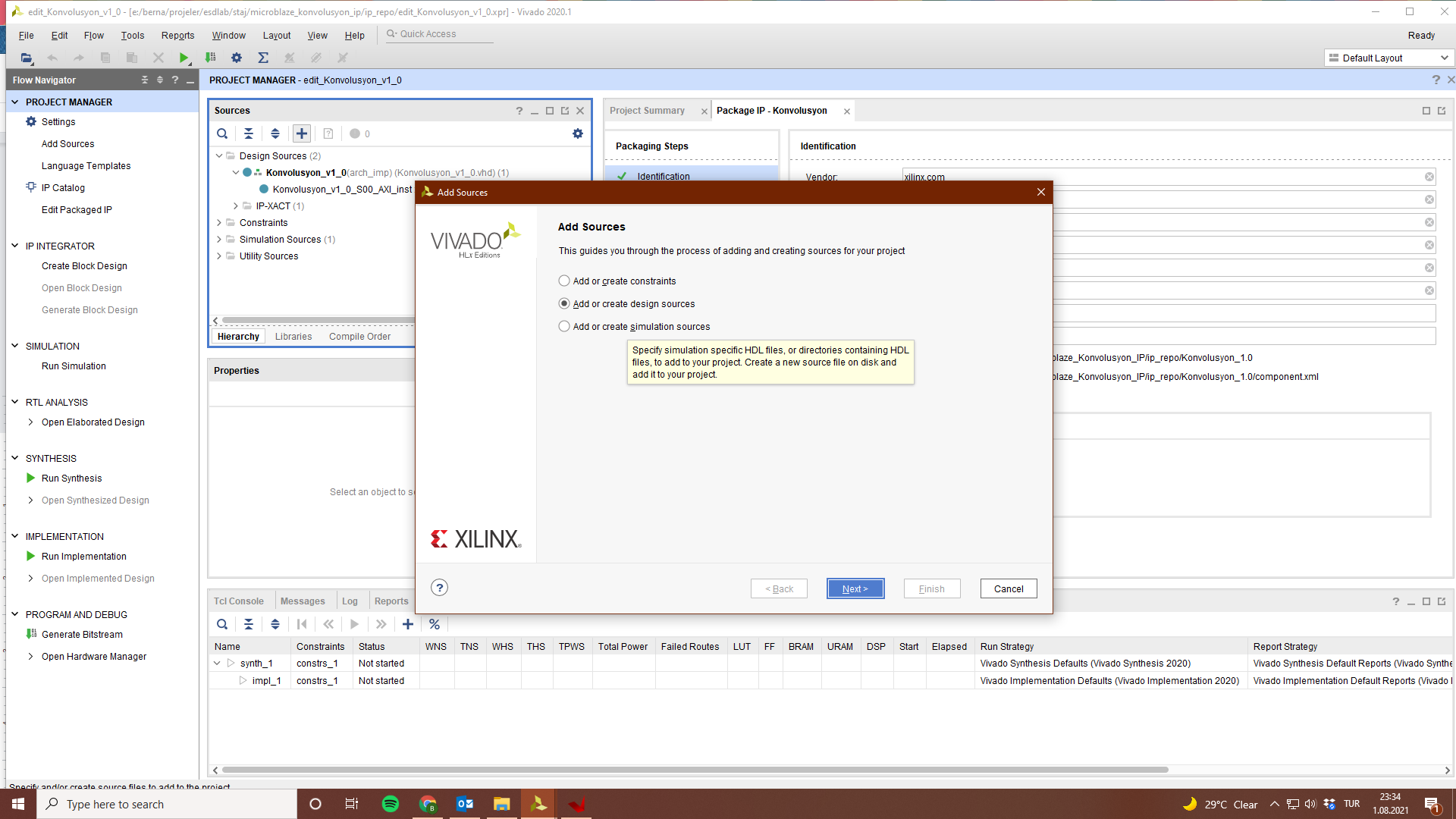


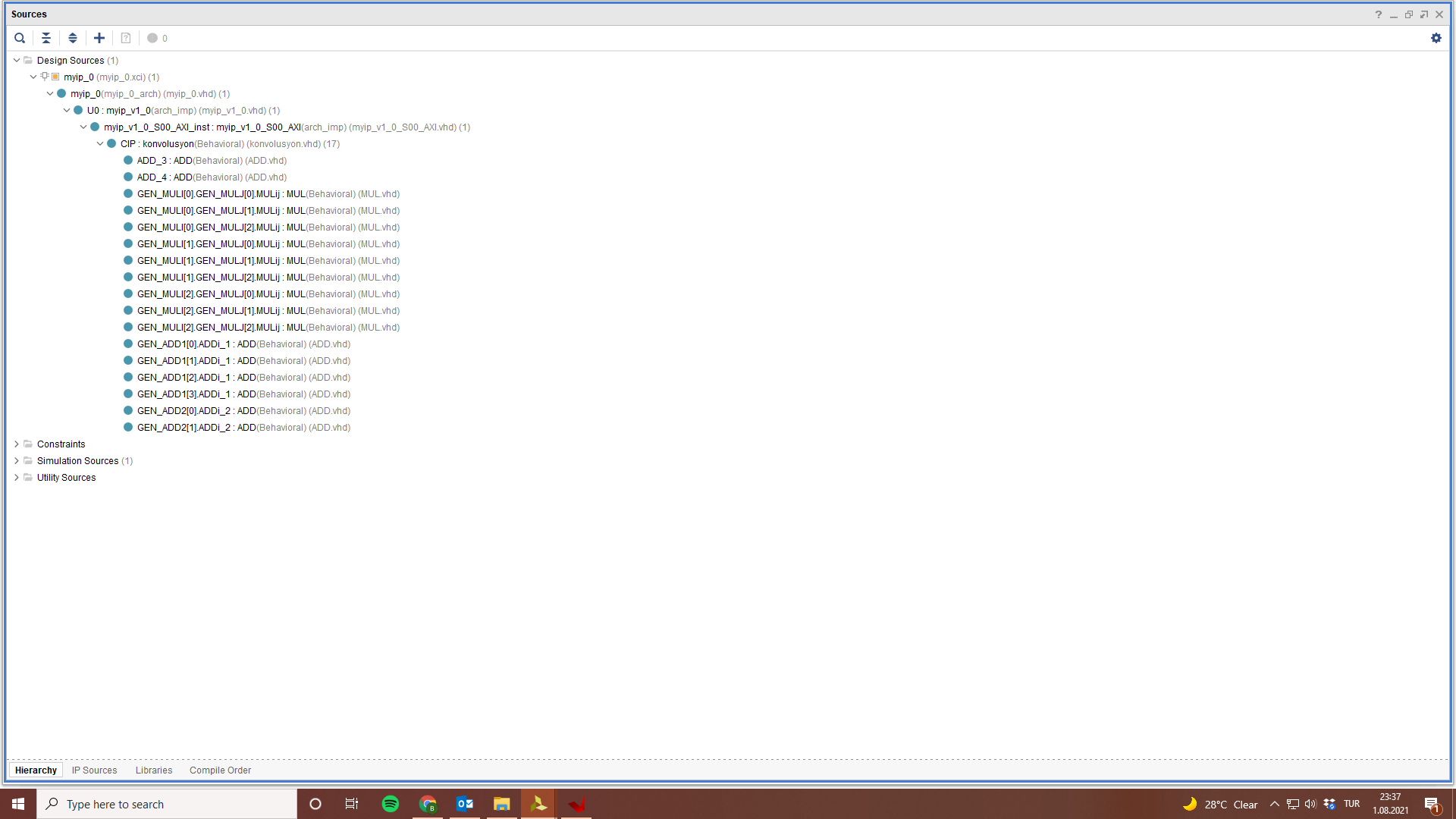




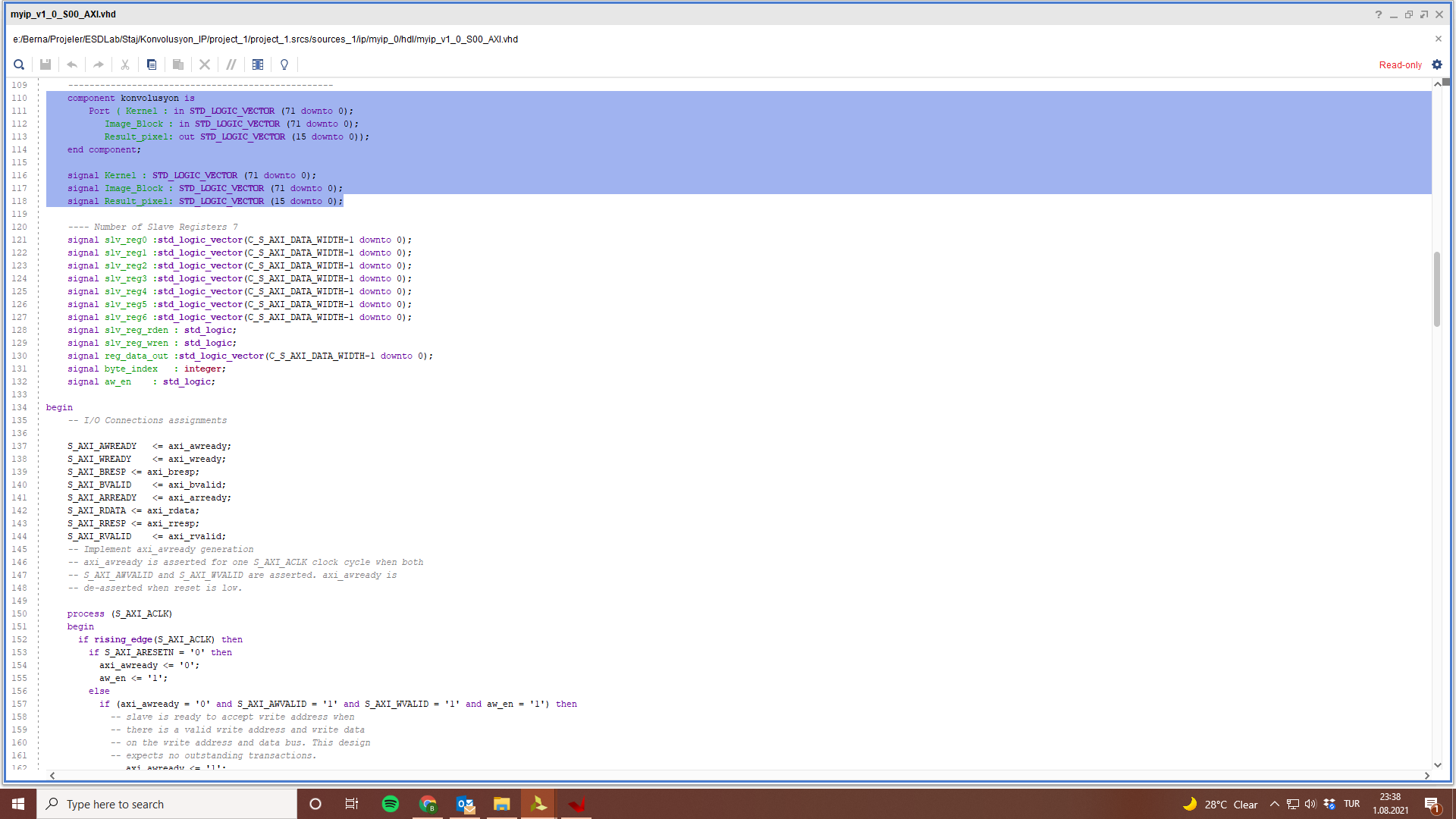






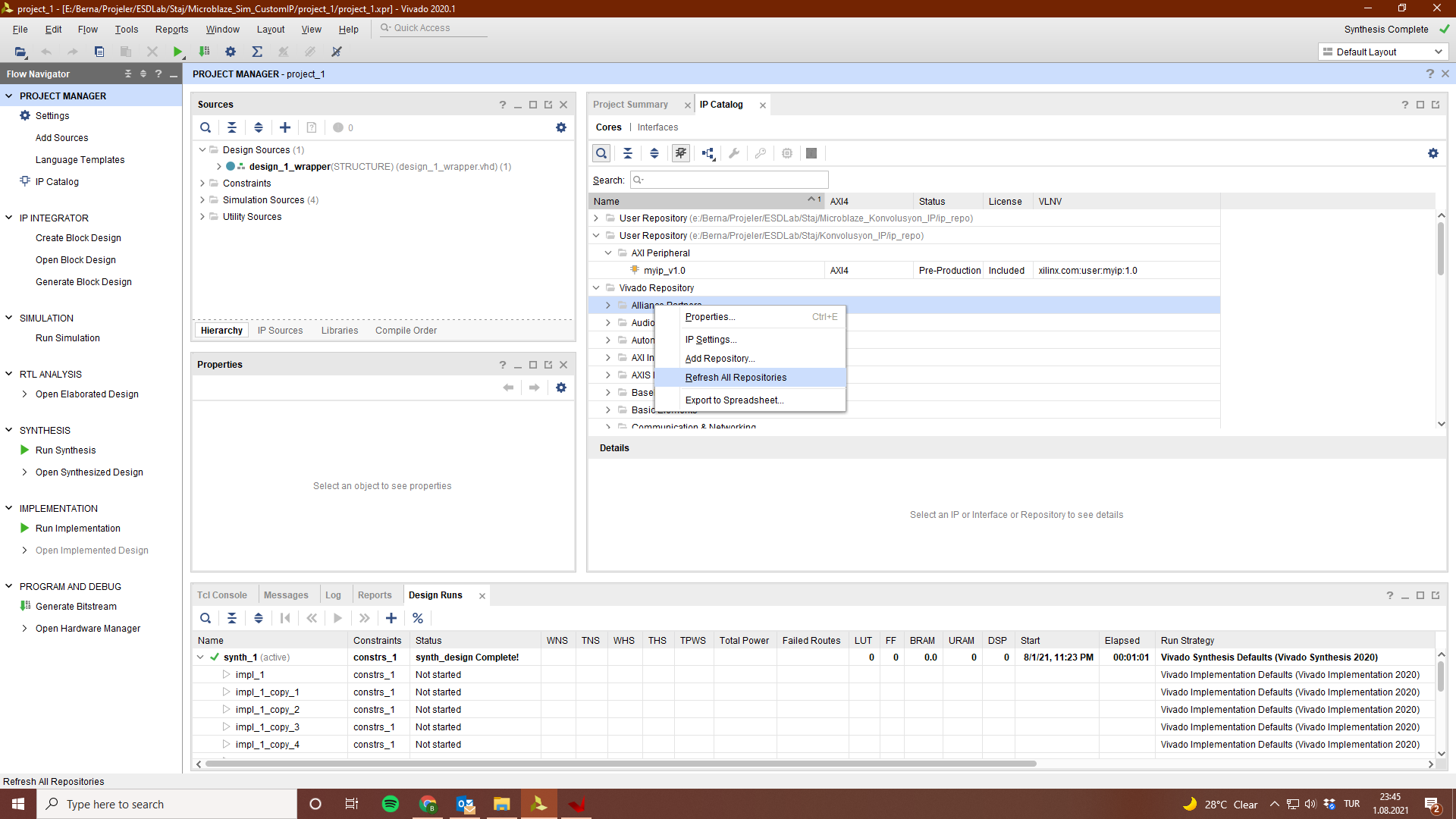


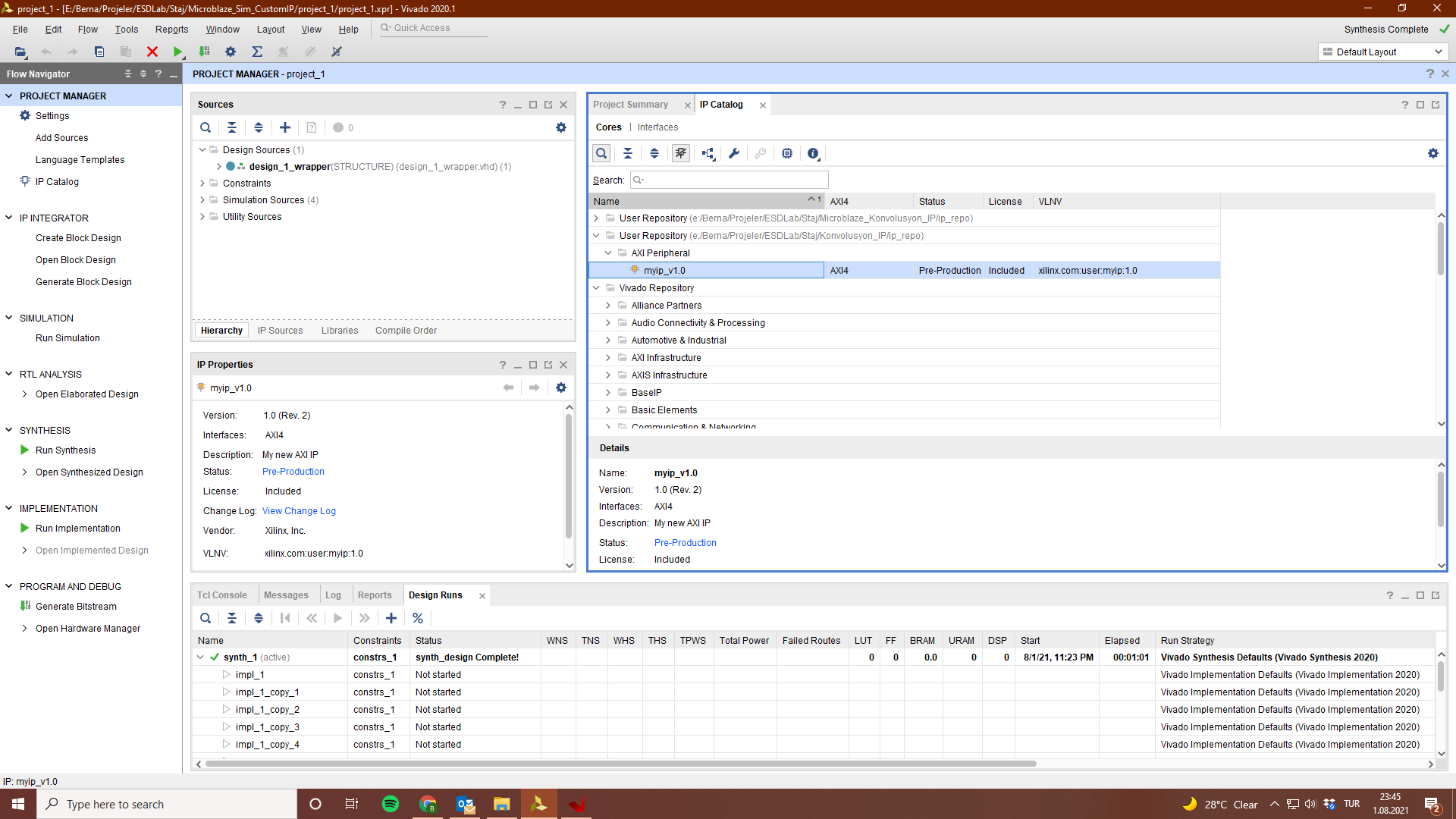
44) AXI bağlantısını tanımlayan dosyanın içine konvolusyon bloğunu koyun.

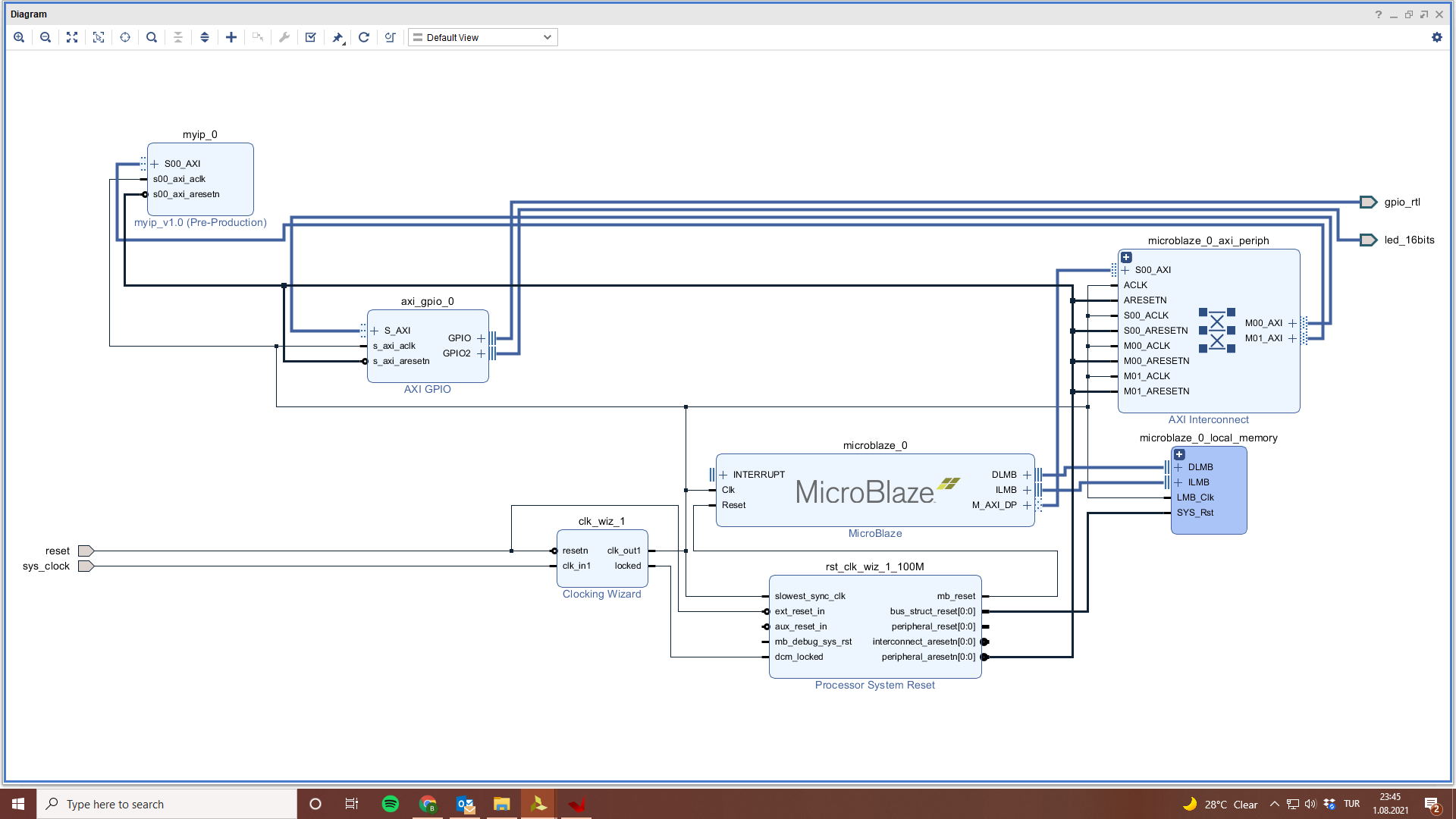




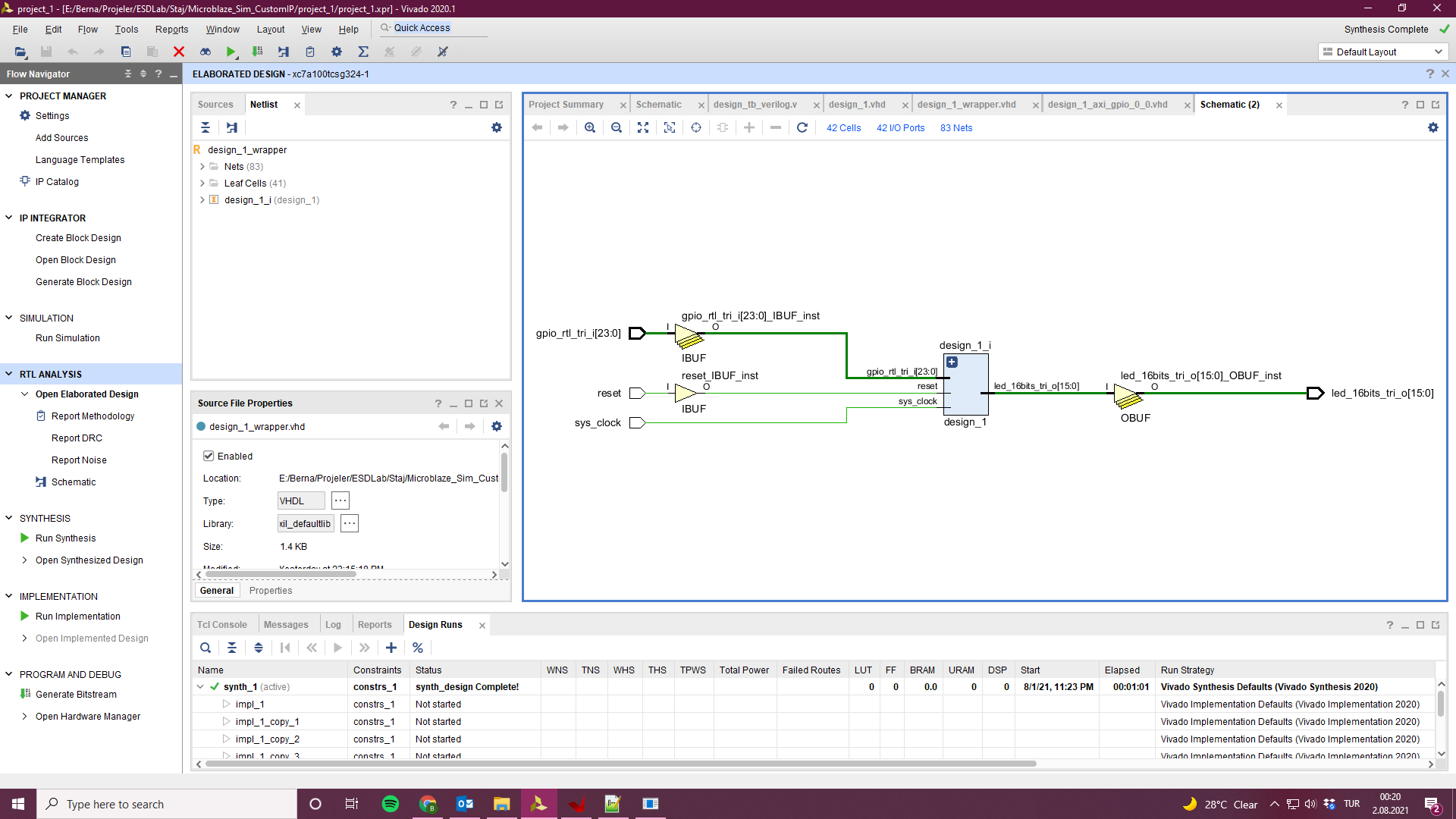
45) Yeni bir Microblaze projesi oluşturup konvolusyon IP sini ekleyelim.

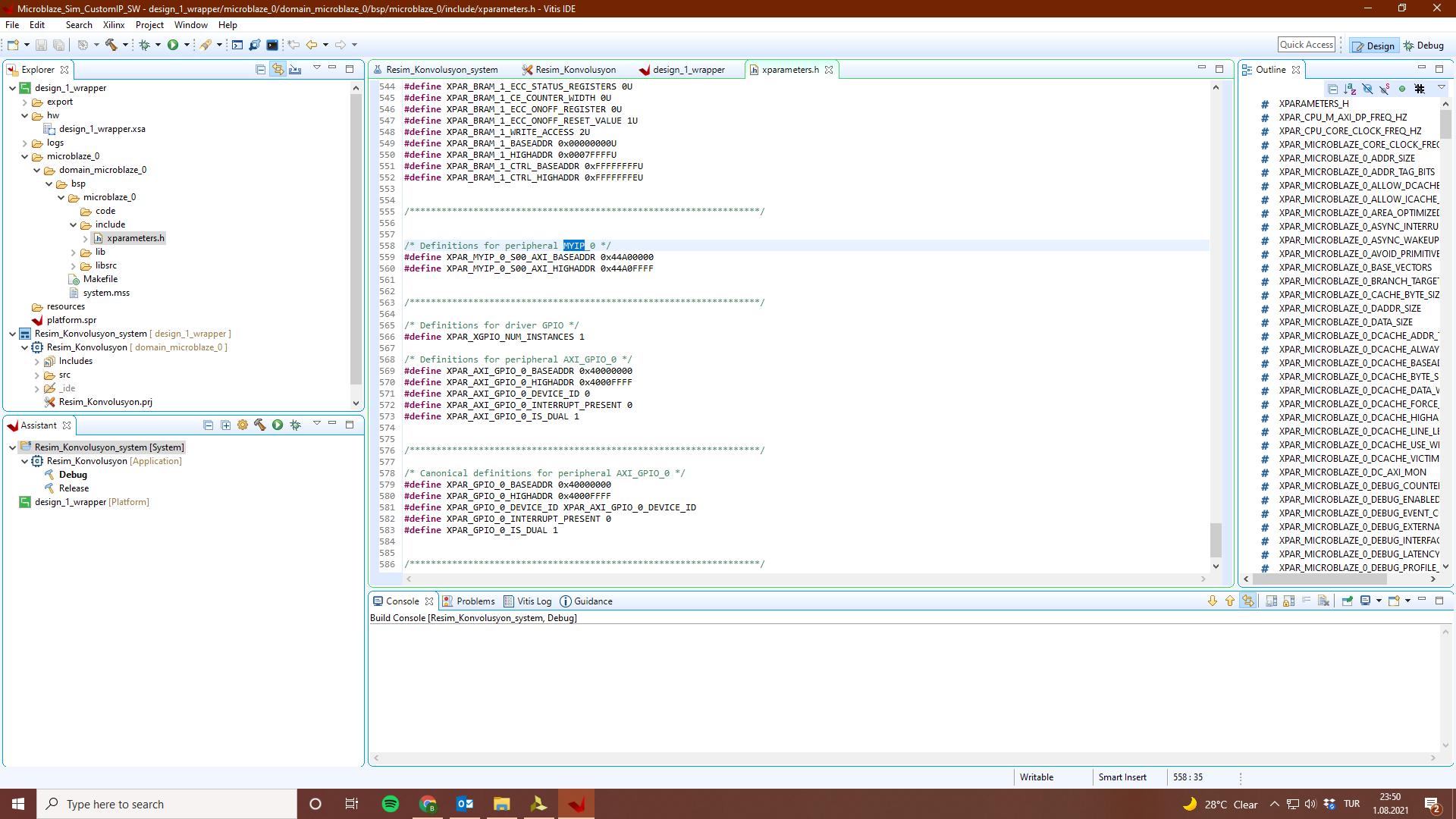


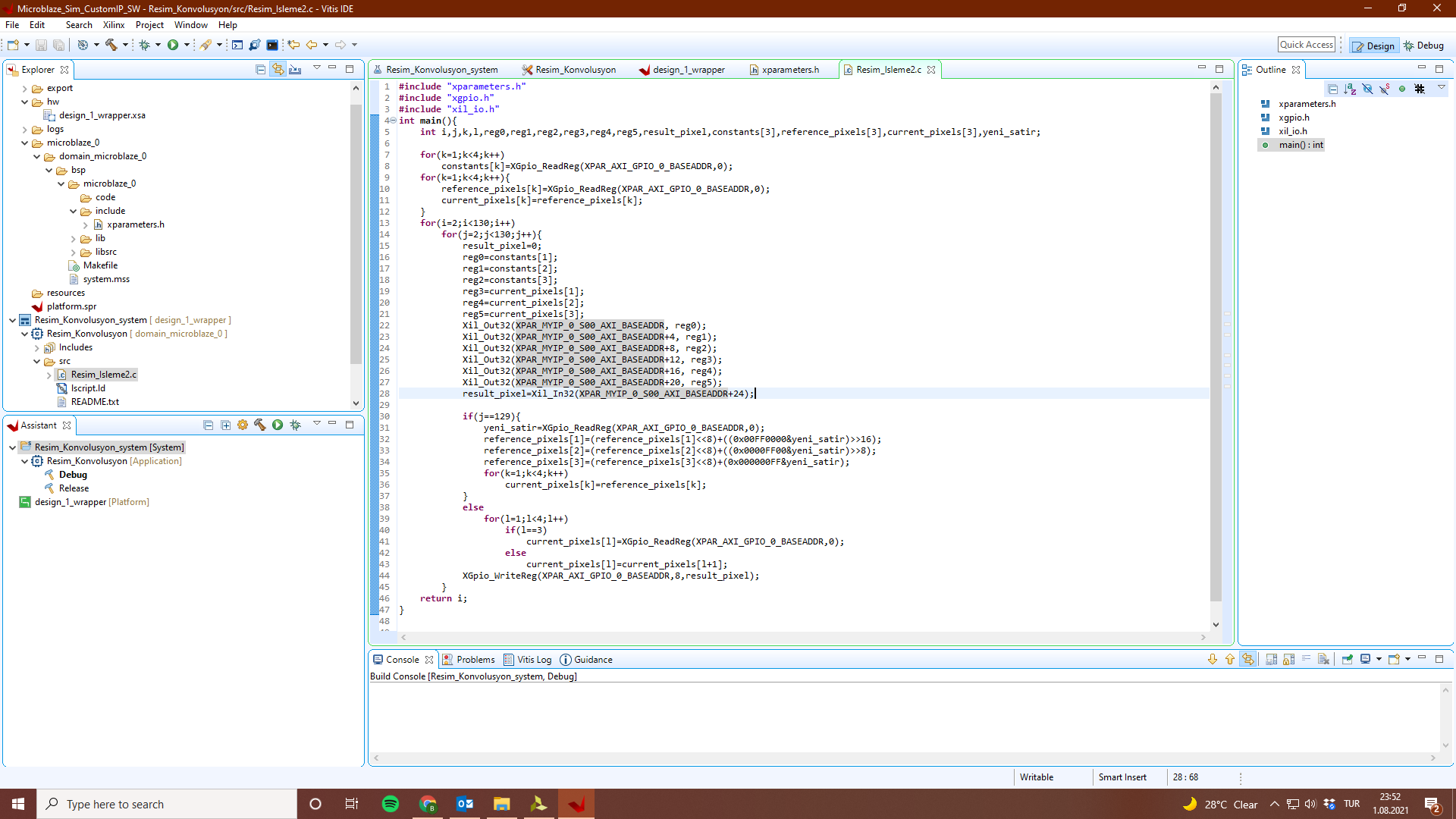




46) Create HDL wrapper, synthesis, export hardware, launch vitis







47) Testbench yazın.

`timescale 10ns / 1ns

module design\_tb\_verilog;

reg [23:0] gpio\_rtl\_tri\_i;

wire [15:0] led\_16bits\_tri\_o;

reg reset;

reg sys\_clock = 0;

integer i,j,k,l;

integer fin\_pointer,fout\_pointer;

reg [7:0] input\_image[1:130][1:130];

design\_1\_wrapper UUT (.gpio\_rtl\_tri\_i(gpio\_rtl\_tri\_i),.led\_16bits\_tri\_o(led\_16bits\_tri\_o),.reset(reset),.sys\_clock(sys\_clock));

always #5 sys\_clock = ~sys\_clock;

initial

begin

fin\_pointer= $fopen("input\_image.txt","r");

for(i=1;i<131;i=i+1) begin

for(j=1;j<131;j=j+1) begin

$fscanf(fin\_pointer,"%d\n",input\_image[i][j]);

end

end

$fclose(fin\_pointer);

fout\_pointer= $fopen("output\_image.txt","w");

reset = 0;

# 10;

reset = 1;

//filter constants are being written

gpio\_rtl\_tri\_i[23:16] = 1;

gpio\_rtl\_tri\_i[15:8] = 2;

gpio\_rtl\_tri\_i[7:0] = 1;

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_rdack\_i);

gpio\_rtl\_tri\_i[23:16] = 2;

gpio\_rtl\_tri\_i[15:8] = 4;

gpio\_rtl\_tri\_i[7:0] = 2;

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_rdack\_i);

gpio\_rtl\_tri\_i[23:16] = 1;

gpio\_rtl\_tri\_i[15:8] = 2;

gpio\_rtl\_tri\_i[7:0] = 1;

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_rdack\_i);

for(l=-1;l<2;l=l+1) begin

gpio\_rtl\_tri\_i[23:16] = input\_image[2-1][2+l];

gpio\_rtl\_tri\_i[15:8] = input\_image[2][2+l];

gpio\_rtl\_tri\_i[7:0] = input\_image[2+1][2+l];

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_rdack\_i);

end

for(i=2;i<130;i=i+1)

for(j=2;j<130;j=j+1) begin

if(j==129) begin

gpio\_rtl\_tri\_i[23:16] = input\_image[i+2][2-1];

gpio\_rtl\_tri\_i[15:8] = input\_image[i+2][2];

gpio\_rtl\_tri\_i[7:0] = input\_image[i+2][2+1];

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_rdack\_i);

end else begin

gpio\_rtl\_tri\_i[23:16] = input\_image[i-1][j+2];

gpio\_rtl\_tri\_i[15:8] = input\_image[i][j+2];

gpio\_rtl\_tri\_i[7:0] = input\_image[i+1][j+2];

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_rdack\_i);

end

@(posedge UUT.design\_1\_i.axi\_gpio\_0.U0.ip2bus\_wrack\_i);

$fdisplay(fout\_pointer,"%d",led\_16bits\_tri\_o/16);

end

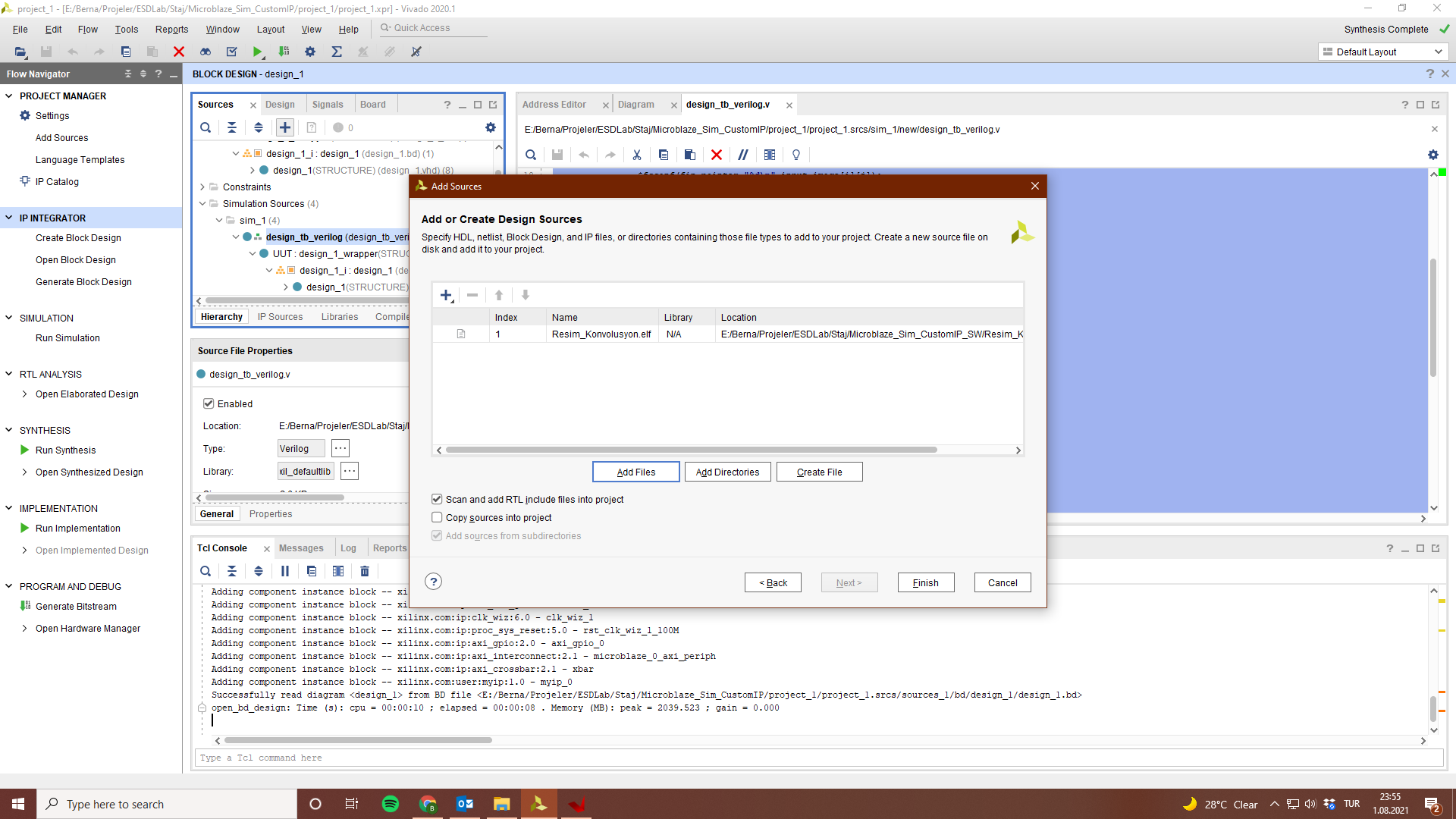
$fclose(fout\_pointer);

$finish();

end

endmodule

48) \*.elf dosyası eklenecek. Associate elf file



49) Simulation