

# **GRABAGAILUA**

Sistema Digitalen Diseinua eta Eraikuntza

EGILEAK: Aitor Madrazo  
Xabier Peñagarikano

# SARRERA

Proiektu honen helburua sistema digitaletan gehiago murgiltzea da, eta horretarako grabagailu bat eraiki beharko dugu geure diseinuak erabiliz.

Grabagailua plaka abtean oinarritzen da mikrofono bat eta bozgorailuak konektatuak dituelarik. Erabiltzailea, Mikronotik soinua grabatu ahal izango du eta hau plakan gorde. Ondoren, grabatutakoa erreproduzitzeko aukera izango du.

Adierazitako plakak hainbat osagai ditu eta guk hauetako batzuk erabiliko ditugu haien funtzionamendua programatuz.

# TRESNAK

Aurretik aipatutako gailua “Altera” etxeko DE1-SoC plaka izango da, bertara mikrofono sarrera bat eta estereo TRS motako (mini jack 3.5mm) irteera konektatua. (Edozein)

Hardwarea programatzeko, ModelSim\*-Intel® FPGA Starter Edition programa erabiliko dugu vhdl-ak (.vhd formatua) sortuz.

Bertan, hardwarearen funtzionamendua erazagutu eta simulazioak egiteko aukera izango dugu, hala gure diseinuaren funtzionamendua aztertu ahal izango dugu plakan martxan arazoren bat egonez gero oso zaila bait da arazoa zein den atzematea.

Diseinua martxan jarri eta probatu ahal izateko, Quartus Prime Lite edition, 18.1 programa erabiliko da. Honekin, “Altera” etxeko gailuetan martxan jarri daitezke aurretik “Modelsim” programarekin egindakoa, eta denbora errealean probatu. Hala ere, konplexuagoa da bertan arazoak atzematea, beraz garrantzitsua da aurretik diseinu sendo bat izatea simulazioen bidez ondo probatua.

# PROZESUA

Lehenik proiektu osoa “fase” ezberdinetan banatzen da, kontuan izanik grabagailua egiteko plakaren zenbait gailu beharko direla bakoitza funtzio ezberdin batekin, lehenik gailu ezberdinetan banatuko da prozesua. Gailuak banatuta ditugula, hauen funtzionamendua ulertu eta eta bakoitzak izango duen funtzioa zehaztu behar da. Ondoren, papelean bakoitzaren diseinua egin beharko da, bai prozesu unitatea bai kontrol unitatea. Diseinua prest dagoelarik, diseinu honen vhdl-a sortuko dugu “Modelsim” programan.

Hona iritsita, garrantzitsua da simulazioan dena ondo joatea. Horretarako, “Modelsim”-en gailua konpilatu eta errore guztiak zuzendu ondoren, “testbench”-a sortu beharko da, non simulatzerakoan seinale konkretu batzuen hasieraketak egiten diren eta martxan jartzeko portaera zehaztu behar den. Dena konpilatuta eta errorik gabe edukita prest egongo da simulazioa hasteko, nahi diren seinaleak ikusteko da eta zenbat denboran simulatu nahi den adierazi beharko da. Honela ikus dezakegu zein portaera izan dezakeen diseinatutako gailuak, eta ikusitakoarean arabera diseinua aldatu. Oso erabilgarria da fase honetan portaera ondo aztertzea bertan inon baino hobe atzeman bait daitezke egindako akatsak.

Diseinuak banaka oraindik ez daude plakan proban jartzeko prest, ez bait dute ezer egiten bakar lanean, baina esaterako gure kasuan gailu gutxi batzuekin eta grabazio bat izanik, erreproduktore bezala proban jarri daiteke.

Aurrera jarraitzeko, diseinuak "Quartus"-era pasa eta plaka martxan jarri behar dugu. Honela egindakoa praktikan jartzen da denbora errealean probatuz, seinale gutxi batzuk ikuskatzeko aukera ematen du baina ez gehiegi, beraz ez da eraginkorra arazoak edo akatsak aztemateko.

Lehenik "Au\_Out" moduluarekin hasi ginen, zeinak 16 biteko lagin bat hartuta bitez bit pasatzen duen ezkerreko zein eskuineko kanaletarako.

Ondoren, "ramsinc" modula eskuan izanik "main" diseinuarekin jarraitu dugu, memorian gordetakoaren laginak "Au\_Out" moduluari pasatzen zaizkio eta honek azkenik codec-era bidaliko du seinale digitala analogiko bihurtuz eta erreproduzituz. Beraz, honela erreproduktore bat lortzen da, aurrerago beste modulu batzuk gehituta grabatzeko aukera ere gehituko da.

## ARAZOAK

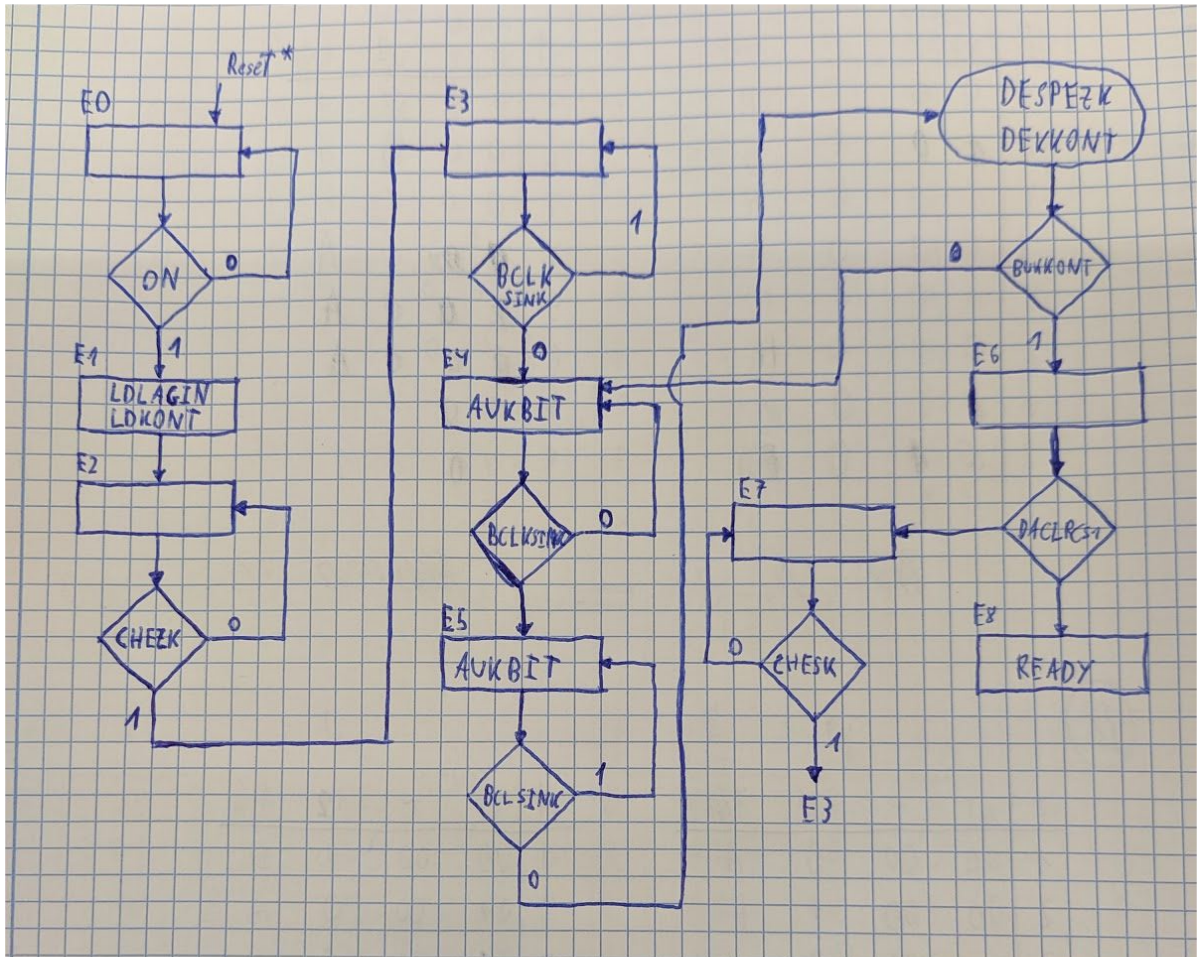
Orokorrean, izan ditugun arazoak lengoaiarekin lotutakoak dira. Baina ingurukoei eta irakasleari galdetuta aurrera egitea posible izan zaigu orain arte.

Zenbait akatsez konturatzeko simulazioa erabilgarria izan da, espero ez genituen portaerak agertu bait zaizkigu, eta hauek zuzentzea batzuetan kostatu egin zaigu.

Oraingoz "main" fasean simulazioan blokeatuta gaude, kontrol unitateko egoera batean "Au\_Out" moduluaren zain gelditzen delarik.

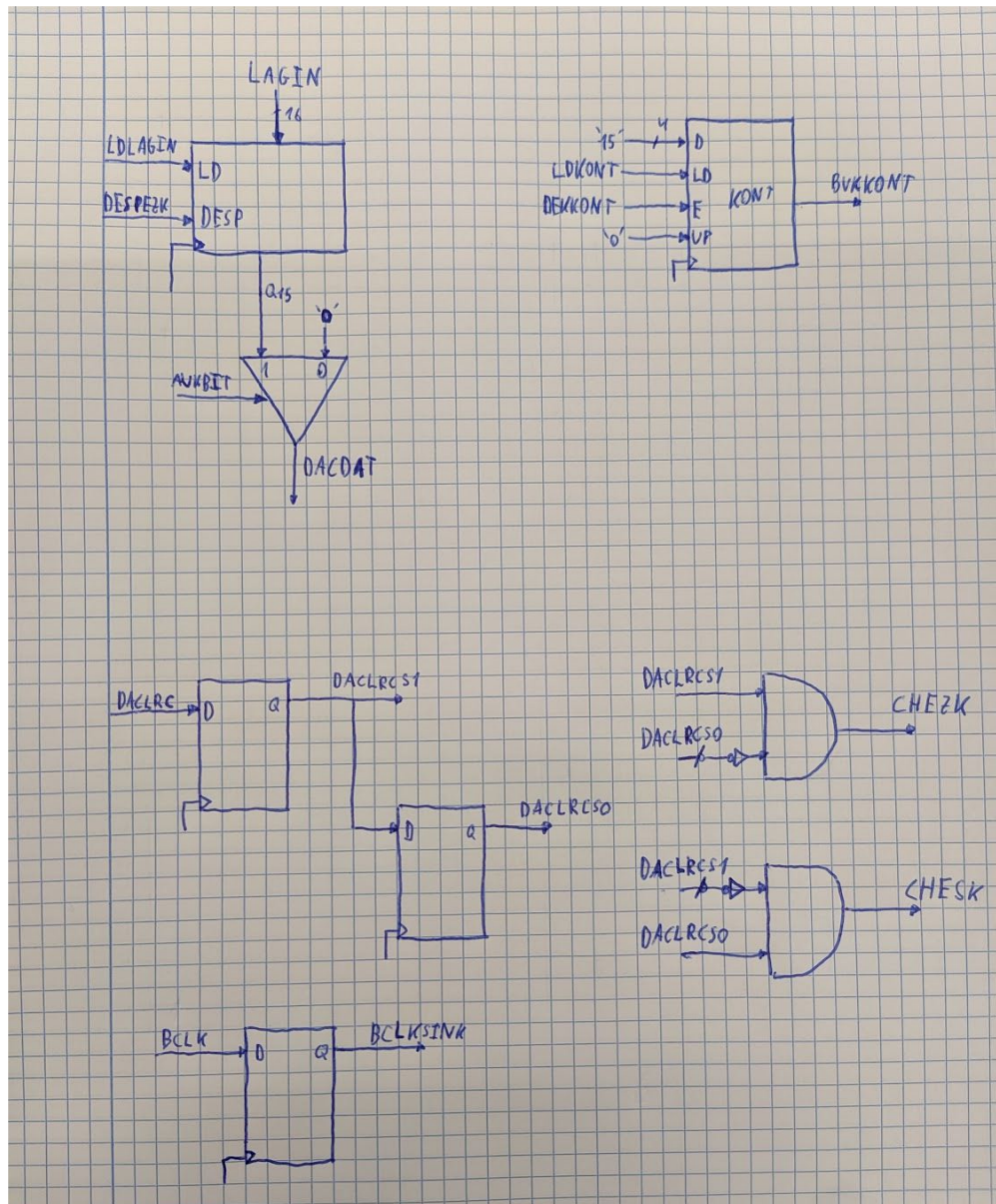
# AU\_OUT GAILUA

## KONTROL UNITATEA



Hasteko labin bat kargatzen da eta honetarako "ON" seinalea aktibatzen denean, lehenengo egoerara pasa eta lagina kargatzen hasteko bi seinaleak aktibatuko dira. Ondoren, ezkerreko kanalaren seinalea aktibatzeko zain geratzen da, eta hau aktibatzean hasiko da prozesatzen bitez bit. BCLKSINK seinaleak erloju antzera funtzionatu du bita kargatzeko sinkronizaturik. BUKKONT seinaleak adieraziko du bit guztiak prozesatu direla, eta beraz, eskuineko kanalarekin hasi behar dela. Baina seinale hau aktibatzean ez badago eskuineko kanalarekin jarraitzerik (hau DACLRCS1-ek adieraziko du) hurrengo labinaren zain geratuko da.

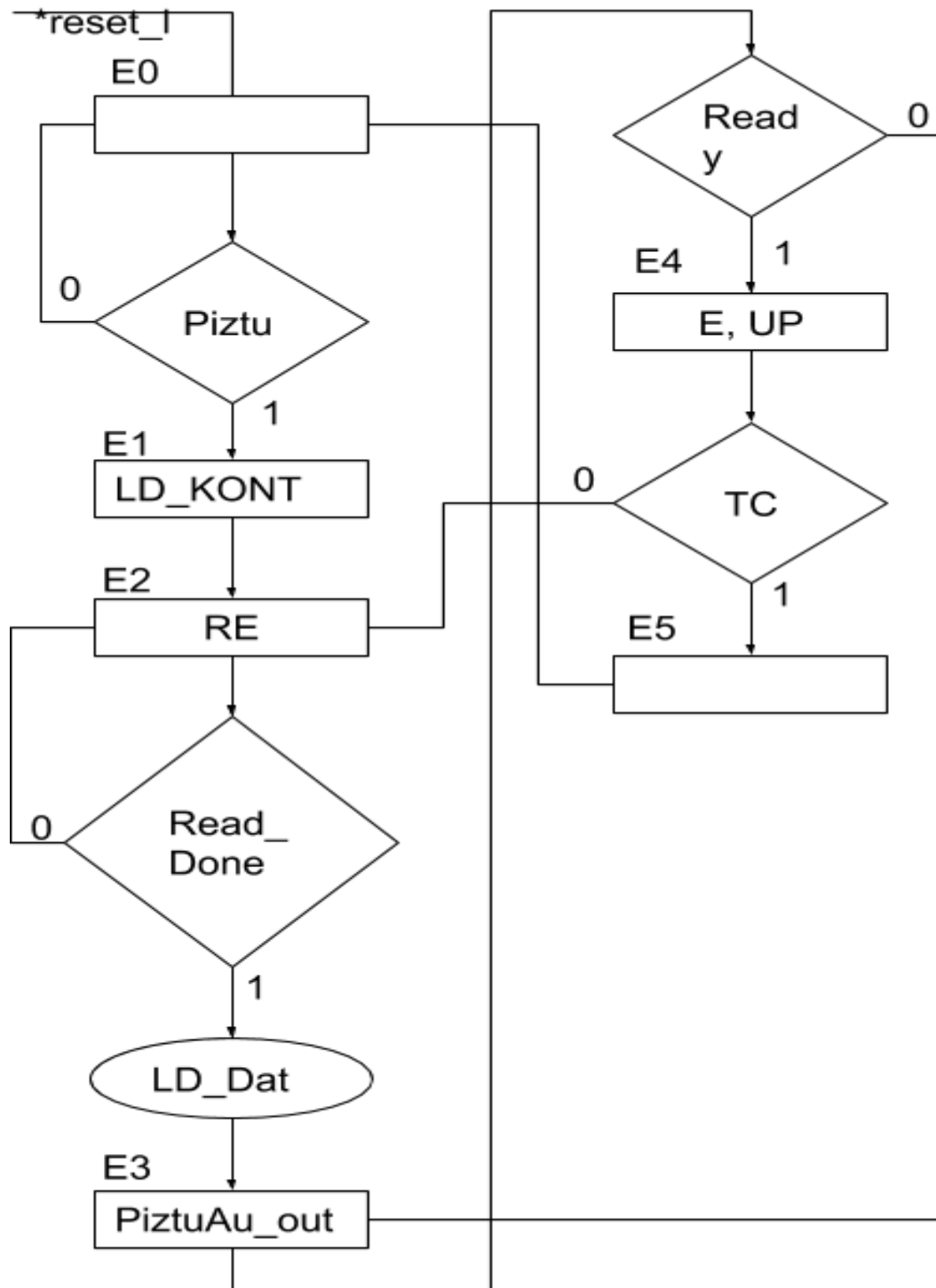
# PROZESU UNITATEA



- Desplazamendurako gailuak lagina jasotzen du eta bitez bit ematen du emaitza moduan ziklo bakoitzeko, lehen aipatutako moduan kontrol unitateko DESPEZK seinala bita aukeratzeko lagina desplazatzen joango delarik.
- Kontagailua laginaren 16 bitak kontatzeaz arduratzen da, 16 bit hauekin bukatzean, kontrol unitateari abixua pasako bait dio kanalez aldatu edo amaitu dezan.
- Gainontzeko egiturak sinkronizazioa mantentzeko dira. BCLKSINK seinala erloju antzera erabiliko dena, eta beste biak DACLRC seinalaren arabera ezkerreko zein eskuineko kanalerara aldatzeko.

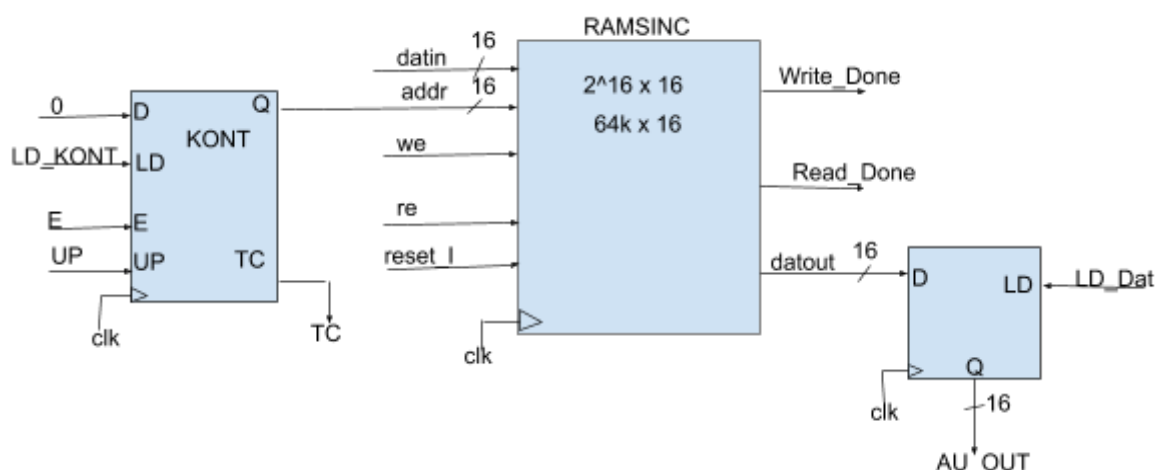
# MAIN

## KONTROL UNITEA



Hasieran, lehenengo egoeran egongo gara “Piztu” botoiari itxaroten. Pizteko botoia sakatzean hurrengo egoerara pasatuko gara, “LD\_KONT” aktibatu eta 16 bitak kontagailuan kargatuko dira. Erlojuaren maiztasunarekin hurrengo egoerara pasa eta “RE” aktibatuko da. “addr”-tik sartuko dira 16 bitak RAMSINC-era. Orduan datout iteratik joango dira 16 bitak. “RE” aktibaturik egongo da “Read\_Done” bat-era jarri arte, hau da irakurri arte. “Read\_Done” aktibatzean LD\_Dat momentu baten piztu eta erregistroan gordetako 16 bitak erregistrora joango dira. “AU\_OUT”-eko kontrol unitateko “PiztuAu\_out zai zegoena aktibatuko eta “AU\_OUT”-en kontrolak bukatu arte zai egongo gara, hau da Ready batera jarri arte itxarongo dugu. Orduan lehenengo 16 bitekin bukatuko dugu. Orain kontagailuaren “E” eta “UP” piztuko dira eta berriro ere aurreko prozesu guztia egingo du kontagailuko “TC” irteera aktibatu arte. Hasierako egoerara bueltatuko gara berriro ere “Piztu” botoia sakatu arte itxarotera. Edozein momentutan “reset\_l” sakatzen bada, hau da reset\_l zerora pasatzen bada hasierako egoerara pasatuko da.

## PROZESU UNITATEA



- Erregistroak “datout”-tetik datorren datua gordeko du LD\_Dat aktibatzean, eta ondoren Au\_Out-i pasako dio datua.
- Kontagailua 0 helbidetik hasi eta banan banan igotzen joango da helbide guztiak pasatu arte. 16 bit hauekin bukatzean, “TC” aktibatuko da eta prozesua hasieratik hasiko da berriro ere.
- RAMSINC en ardura “addr”tik helbidea jasotzean “RE” aktibatu eta “datout” irteeratik bidaltzea da. Datua pasatu bada “Read\_Done”-ek datua irakurriko du eta bateko bat bueltatuko du, zero ko bat bueltatuko du irakurtzen ez duen bitartean.

