



Detyra e dyte nga lenda Arkitektura e Kompjutereve

Tema: Dizajnimi i nje CPU 16-biteshe (Single Cycle)

Prof :

Valon Raca

Anetaret e grupit:

Diellza Gllareva-180714100162

Elire Raci -180714100106

Elsa Nishori -180714100063

Hyrje

Ne kemi filluar dizajnimin e nje CPU-je 16 biteshe duke modifikuar nje alu 1-biteshe dhe pastaj duke e kthyer nje alu 32-biteshe ne 16 sipas specifikimeve te nevojshme dhe kemi realizuar disa instruksione qe alu jone i perkrahe si psh: and, sub, add, or, xor, mult, shift left, shift right.

Keto instruksione kemi mundur te i therrasim dhe te i ndajme sipas kerkesave te detyres me op 3 bitesh permes nje multiplekseri 4me1 qe punon ne parimin e operatorit ternar, dhe ne fund testimi eshte bere ne testbench ku edhe i kemi dhene vlera parametrave qe kane qene te domosdoshem per implementimin e ketyre instruksioneve.

Nga kerkesat opsionale ne kemi realizuar xor-in, shift left-in, shift right-in dhe mult-in por multin me ndihmen e disa instruksioneve te gatshme per arsye se kemi pas komplikime ne fund qe ta implementojme mult-in me module te vecanta.

Pastaj kemi realizuar edhe Single Cycle Datapath ne te cilen i kemi instancuar Register File, Data dhe Instruction Memory dhe Alu-ne tone 16-biteshe. Ketu kemi bere lidhjen e ALU-se me Register File, Data Memory dhe Instruction Memory.

Njesine e kontrollit(CU) e kemi pershtatur sipas specifikimeve te dhena ne detyre, kurse ALU Control e kemi realizuar ne ate menyre ku permes ALUOp-it 1-bitesh dihet se cfare operacioni realizohet, kurse permes function tregohet se cili nga instruksionet e ALU-se po realizohet.

Ne fund Datapathin dhe Njesine e Kontrollit I kemi instancuar brenda CPU-se dhe e kemi testuar ne testbench.

Dizajni

File-at tone jane te organizuar ne folderin Kodet.

Tek folderi Kodet ndodhen keta files:

Alu1bit- qe e permбан alu 1 biteshe e cila I perkrahe keto instruksione: add, and, sub, or, xor, mult, shift left, shift right.

Alu16bit- qe e permбан alu 16 biteshe e cila eshte bashkesia e 16 alu-ve 1 biteshe.

Mux4to1- qe e permбан multiplekserin ne te cilin thirren instruksionet ne baze te opcode-it specifik.

Shiftright- ku ndodhet moduli ne te cilin realizohet instruksioni shiftright,

Shiftright- ku ndodhet moduli ne te cilin realizohet instruksioni shiftright,

Multiply- ku ndodhet moduli ne te cilin realizohet instruksioni multiply,

Datapath- ku instancohen file-at e tjere dhe lidhen ndermjet vete,

Register File- i cili i percakton regjistrat RS,RT,RD,

Alu Control- i cili tregon se cili operacion do te ekzekutohet ne baze te AluOp-it ose cili instruksion ne baze te funksionit,

FullAdder- i cili eshte modul ku realizohet mbledhesi i plote,

Mux2to1- multiplekser qe na duhet per bnegate,

File-at per memorie:

Data Memory- i cili eshte modul Read-Write,

Instruction Memory- e cila eshte vetem Read-Only

Cpu- i cili eshte moduli ku instancohet Datapath-i dhe Njesia e kontrollit.

Ekzekutimi

File-at per ekzekutim jane te vendosur brenda folderit Testbench.

Folderi Testbench i permbane keta files:

Alu1bit_testbench- ku behet testimi i alu-se 1biteshe. Ne rastin kur e bejme Run Simulation do shfaqet nje dritare ku kemi mundesi te shohim realizimin e instruksioneve te ndryshme te cilat i perkrahe alu-ja 1biteshe varesisht opcodit dhe rezultatet qe jep secili instruksion.

Alu16bit_testbench- ku behet testimi i alu-se 16biteshe. Ne dalje do fitojme pamjen e instruksioneve qe i perkrahe alu-ja jone, por kjo nuk eshte 1biteshe si alu-ja paraprake por eshte 16 biteshe.

Cpu_testbench- ku behet testimi i Cpu-se 16biteshe. Ne dalje fitohet nje clock 0,1,0,1,..., e cila ka funksionuar vetem pasi qe kemi pare se datapath-i dhe njesia e kontrollit jane realizuar paraprakisht.

File-s tjere per testim nuk kemi ne dorzimin final,pervec atyre per data memory,instruction memory dhe register file, qe i kemi marre te gatshme nga ju, por jemi siguruar qe operacionet e ndryshme si psh: shift left, shift right, multiply t'i testojme me testbench te krijuar vetem per ato module dhe pasi qe e kemi pare se po funksionojne testimin e tyre e kemi futur ne alu1bit_testbench dhe alu16bit_testbench.

Përfundimi

Ky projekt ka qene projekti me sfidues deri me tani. Cdo dite pune ne kete projekt ka pasur sfidat e veta, testime te shumta, hulumtime poashtu.

Besojme qe ky projekt edhe pse ka qene mjaft e veshtire per t'u realizuar, edhe pse nuk kemi mundur ta realizojme deri ne fund, ka qene projekti qe per shkak se ka kerkuar nje perkushtim shume te madh na ka pergaditur per lenden e Arkitektures se Kompjutereve shume me shume sesa vetem leximi i literatures se lendes.

Ne kemi arritur te plotesojme opsionet : a, b, c si dhe disa instruksione nga fusha opsionale g.

Edhe pse ndoshta projekti nuk eshte i realizuar ne menyren me te mire te mundshme dhe nuk i kemi plotesuar te gjitha kerkesat, ky projekt na ka mesuar virtyte qe nje inxhinierit asnjehere nuk duhet ti mungojne : te jesh i durueshem dhe te mos dorzohesh pasi qe shume nga modulet kane pas rezultat pas dhjetra testimesh.