ARKITEKTURA E KOMPJUTERËVE

Viti Akademik 2019/2020

DETYRA 2

Valon Raca & Vlera Alimehaj

1. Hyrje

Sipas syllabusit të lëndës Arkitektura e Kompjuterëve vlerësimi i studentëve është i përzier. Nga 100% të pikëve që formojnë notën përfundimtare, vlerësimi i vazhdueshëm merr pjesë me 40% dhe testi përfundimtar me 60%.

Detyra 2 vlerësohet me maksimalisht 30% (40% bashkë me bonus) dhe është detyra e fundit semestrale.

Detyra dorëzohet në formën e kodit dhe raportit.

2. Formimi i grupit

Grupi në përbërje prej maksimalisht tre anëtarëve duhet të krijohet më së largu deri me 01.05.2020 në ora 13.30. Për të deklaruar grupin luteni që të shënoheni emrat e anëtarëve të grupit në formën e krijuar enkas në http://www.kompjuterika.tk me emrin "Forma per deklarimin e grupeve per detyren e dyte" që është vendosur në ballinën e lëndës Arkitektura e Kompjuterëve apo që u është shpërndarë përmes email-it.

Studentët kanë të drejtë të mos formojnë grupe dhe ta dorëzojnë detyrën individualisht. Gjithsesi, edhe në këto raste duhet ta shënojnë emrin në formën e grupeve.

3. Detyra

Detyra është e ndarë në pjesë që mbivendosen njëra mbi tjetrën (hollësitë në seksionin 3.3 të këtij dokumenti).

Detyra e juaj është dizajnimi i një CPU 16-bitëshe (Single-Cycle).

3.1. Materiali bazë

1. Si pikë startuese e këtij projekti duhet të jetë dizajni i ALU-së 32 bitëshe që e kemi bërë në Javën e II (Shtojca B.5 e librit kryesor, Ushtrimet në Verilog: Java IX). Për dizajnimin e funksioneve nuk mund të përdoren rrugë të shkurtra në Verilog si p.sh. A + B, për mbledhësin. Për secilin funksion duhet krijuar moduli i posaçëm strukturor apo behavior-ist.

2. Si bazë për kontroll të përdoret Njësia e Kontrollit të cilën e kemi punuar në Javën e IX në ushtrime.
3. Register File, Data Memory dhe Instruction Memory do t'iu jepen të gatshme në javën e X.

3.2. Specifikimi i sistemit

CPU duhet të jetë 16 bitëshe (word-in për këtë detyrë do ta konsiderojmë 16 bitësh / 2 bajtësh).

Formati i instruksioneve

CPU duhet të përkrah instruksione të formatit R dhe I.

Formati R

OPCODE		RS			RT			RD			FUNCT				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati I

OPCODE		RS			RT			IMMEDIATE							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ALU

ALU do të përkrah operacione në numra me gjatësi 16 bitëshe.

ALU do të ketë dy hyrje A dhe B, dhe hyrjen CarryIn; një dalje për Rezultat dhe një dalje për CarryOut (Për këtë detyrë nuk kërkohen daljet Zero, Set Less Than, Overflow).

Hyrja B do të ketë përpara një multiplekser që zgjedh në mes regjistrit RT apo vlerës imediate.

Njësia e kontrollit

Nga OPCODE-i dybitësh Njësia Kontrolluese kontrollon dhe drejton njësitë tjera.

Daljet e njësisë së kontrollit:

input	**input.gjatësia[bit]**	**output**	**output.gjatësia [bit]**
OPCODE	2	RegDst	1
		ALUSrc	1
		MemToReg	1
		RegWrite	1
		MemRead	1
		MemWrite	1
		ALUOp	1

ALU Control

Hyrjet

Në hyrje kemi 5 bit nga fusha FUNCT e instruksionit dhe 1 bit nga ALUOp.

ALUOp përcakton se për çfarë kryhet operacioni:

ALUOp	Selekto	Arsyeja
0	Mbledhje	LW apo SW
1	Sipas FUNCT	Operacionet ALU

Një dekoder do të merr hyrjet dhe do t'i shndërroj në dalje sipas specifikimit mëposhtë.

Daljet

Një linjë për invertimin e hyrjes B në ALU - BNegate. BNegate bëhet 1 vetëm në rast të zbritjes dhe shërben për invertimin e B-së dhe për furnizimin e CarryIn me 1.

Tre bitat tjerë shërbejnë për multiplekserin që zgjedh operacionin.

BNegate	Bit1	Bit2	Bit3	Operacioni
0	0	0	0	AND
0	0	1	0	OR
0	0	1	1	XOR
0	1	0	0	ADD
0	1	0	1	ADDI
1	1	0	0	SUB
0	0	0	1	SLL
0	1	1	0	SRL
0	1	1	1	MULT

Register File

Numri i regjistrave do të jetë 8 (regjistri $zero dhe 7 të tjerë për përdorim të përgjithshëm). Regjistrat do të jenë të gjerë 16 bit.

Register File ka tre hyrje tre bitëshe për përcaktimin e regjistrave RS, RT, RD.

Register file ka një hyrje 16 bitëshe për të shkruar në regjistrin RD.

Register File ka dy dalje 16 bitëshe për të lexuar të dhënat nga regjistrat e përcaktuar në RS dhe RD.

Adresat e regjistrave:

$zero	000

$r1	001
$r2	010
$r3	011
$r4	100
$r5	101
$r6	110
$r7	111

Regjistri PC po ashtu është 16 bitësh dhe është i ndarë nga Register File. Për PC të krijohet mbledhës i veçantë.

Memoria

Dy memorie nga 128 bajt secila (Instruction dhe Data Memory)

Harta e memories:

Adresa	Segmenti
0 – 10	E rezervuar
10 - 127	Instruction apo Data

Instruction Memory është vetëm Read-Only. Hyrje 16 bitëshe nga PC për përcaktimin e adresës së instruksionit. Dalje 16 bitëshe për leximin e instruksionit.

Data Memory është Read-Write. Hyrje 16 bitëshe për përcaktimin e adresës së fjalës 2 bajtëshe që lexohet/shkruhet. Hyrje 16 bitëshe për fjalën që shkruhet. Dalje 16 bitëshe për fjalën që lexohet.

Instruksionet

AND

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	X	X	X	X	X	X	0	0	0	0	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: AND rd, rs, rt

Shembull: AND $r1, $r2, $r3

Kryhet DHE logjike bit për bit ndërmjet regjistrit $r2 dhe $r3. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

OR

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	X	X	X	X	X	X	0	0	0	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: OR rd, rs, rt

Shembull: OR $r1, $r2, $r3

Kryhet OSE logjike bit për bit ndërmjet regjistrit $r2 dhe $r3. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

XOR

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	X	X	X	X	X	X	0	0	0	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: XOR rd, rs, rt

Shembull: XOR $r1, $r2, $r3

Kryhet EKSKLUZIV-OSE logjike bit për bit ndërmjet regjistrit $r2 dhe $r3. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

ADD

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	X	X	X	X	X	X	0	0	1	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: ADD rd, rs, rt

Shembull: ADD $r1, $r2, $r3

Kryhet mbledhje ndërmjet vlerave të regjistrit $r2 dhe $r3. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

ADDI

OPCODE		RS			RT			IMMEDIATE							
0	1	X	X	X	X	X	X	Y	Y	Y	Y	Y	Y	Y	Y
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: I

Shablloni: ADDI rt, rs, immediate

Shembull: ADDI $r1, $r2, 100

Kryhet mbledhje ndërmjet vlerës së regjistrit $r2 dhe vlerës imediate 100 (shembull). Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
0	1	0	1	0	0	1

SUB

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	X	X	X	X	X	X	0	0	1	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: SUB rd, rs, rt

Shembull: SUB $r1, $r2, $r3

Kryhet zbritje ndërmjet vlerave të regjistrit $r2 dhe $r3. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

SLL

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	0	0	0	X	X	X	0	1	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: SLL rd, rs

Shembull: SLL $r1, $r2

Kryhet zhvendosje majtas për 1 bit në vlerën e regjistrit $r2. Biti i fundit (LSB) bëhet zero. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

SRL

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	0	0	0	X	X	X	0	1	0	0	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: SRL rd, rs

Shembull: SRL $r1, $r2

Kryhet zhvendosje djathtas për 1 bit në vlerën e regjistrit $r2. Biti i parë (MSB) bëhet zero. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

MULT

OPCODE		RS			RT			RD			FUNCT				
0	0	X	X	X	X	X	X	X	X	X	1	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: R

Shablloni: MULT rd, rs, rt

Shembull: MULT $r1, $r2, $r3

Kryhet shumëzim ndërmjet vlerave të regjistrit $r2 dhe $r3. Rezultati ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
1	0	0	1	0	0	1

LW

OPCODE		RS			RT			IMMEDIATE							
1	0	X	X	X	X	X	X	Y	Y	Y	Y	Y	Y	Y	Y
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: I

Shablloni: LW rt, (immediate)rs,

Shembull: LW $r1, 0($r2)

Vlera imediate zgjerohet nga 8-bit në 16-bit.

Kryhet mbledhje ndërmjet vlerës së regjistrit $r2 dhe vlerës imediate. Rezultati përcakton adresën e cila shërben si input në memorie. Vlera që lexohet nga ajo adresë ruhet në $r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
0	1	1	1	1	0	0

SW

OPCODE		RS			RT			IMMEDIATE							
1	1	X	X	X	X	X	X	Y	Y	Y	Y	Y	Y	Y	Y
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Formati: I

Shablloni: SW rt, (immediate)rs,

Shembull: SW $r1, 0($r2)

Vlera imediate zgjerohet nga 8-bit në 16-bit.

Kryhet mbledhje ndërmjet vlerës së regjistrit $r2 dhe vlerës imediate. Rezultati përcakton adresën e cila shërben si input në memorie. Vlera që gjendet në $r1 shkruhet në adresën e përcaktuar në memorie.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp
0	1	0	0	0	1	0

3.3. Detyrat dhe pikët përkatëse

Të detyrueshme [max. 30 pikë]

a) [5 pikë] Të dizajnohet ALU 16-bitëshe në Verilog sipas dizajnit nga Java e II-të
   i. Të kthehet nga 32-bitëshe në 16 bitëshe ALU sipas specifikimit më sipër
   ii. Të përkrahet së paku AND, OR, ADD, SUB si operacione të ALU
   iii. Të përdoret Ripple-Carry për mbledhësin

b) [3 pikë] Të krijohet Sigle-Cycle Datapath përmes ndërlidhjes së ALU me Register File, Data & Instruction Memory

c) [3 pikë] Të ndërlidhet njësia e kontrollit me pjesët a) dhe b)
   i. Të shtohet ALU Control sipas dizajnit nga ligjëratat
   ii. Të ndryshohet Njësia e Kontrollit dhe ALU Control varësisht nga instruksionet që shtohen

d) [6 pikë] Instruksionet bazë që duhet të përkrahen nga CPU
   i. add
   ii. and
   iii. or
   iv. sub
   v. lw
   vi. sw

e) [8 pikë] Të ekzekutohet programi i mëposhtëm nga memoria:

```
addi $r2, $zero, 109 //vetem nese eshte implementuar
lw $r1, 20($r2)
sub $r2, $r1, $zero
add $r0, $r1, $2
sll $r0, $r1 //vetem nese eshte implementuar
sw $r0, 0($r2)
```

f) [5 pikë] Raporti përfundimtar

Opsionale [Bonus pikë – max. 10 pikë]

g) [0 - 10 pikë] Të shtohen instruksionet e mëposhtme
   i. [1 pikë] XOR (exclusive OR)
   ii. [2 pikë] ADDI (add immediate)
   iii. [2 pikë] SLL (shift left logical) & SRL (shift right logical)
   iv. [5 pikë] MULT (multiplication)

Detyra duhet të dizajnohet ne Verilog dhe testohet në Vivado (apo ndonjë tjetër softuer).

4. Dorëzimi

Detyra dorëzohet dhe pranohet si e tillë, nëse respektohen rregullat e mëposhtme:

- Afati i fundit për dorëzim është 22.05.2020 23.59.

- Dorëzimi pranohet vetëm përmes platformës Moodle të vendosur në ueb faqen http://www.kompjuterika.tk. Kutia për pranimin e detyrës së dytë është vendosur në javën e trembëdhjetë.

- Detyra dorëzohet nga vetëm njëri student, nëse dy/tre studentë punojnë në grup.

- Detyra duhet të jetë e mbështjellur si .zip apo .rar ne formatin GrupiX.zip (ku X eshte numri i grupit – shih emailin per dorëzim dhe prezantim) dhe të ngarkohet te pjesa File Submission.

- Në kuadër të .zip/.rar fajllit duhet të jenë dy fajlla:
  1. Në një folder me emrin **Kodi** vendosini të gjithë fajllat që përmbajnë kod.
  2. Në një folder me emrin **Testbench** vendosini të gjithë fajllat që përmbajnë kodin testues.
  3. Fajlli **Raporti_GrupiX.pdf** (ku X eshte numri i grupit – shih emailin per dorëzim dhe prezantim) që e përmban përshkrimin e punës tuaj. Më gjerësisht shih seksionin Raporti më poshtë.

5. Sanksionet

Studentët janë përgjegjës individualisht dhe kolektivisht si grup për përmbajtjen e detyrës dhe të raportit.

Rreptësishtë ndalohet kopjimi i detyrës nga studentë/grupe të tjera.

Kopjimi i detyrës do të ndëshkohet me 0 pikë për detyrën e parë dhe mospranim për vlerësim të detyrave të tjera këtë semestër. Po ashtu mësimdhënësi e rezervon të drejtën që ta paraqes studentin në Komision disiplinor/Komision të Etikës në FIEK.

6. Raporti

Raporti duhet të përmbledh përvojën tuaj me këtë detyrë dhe të faktojë observimet tuaja.

Raporti duhet të përmbajë këto seksione:

Emri1 Mbiemri1, nr. i ID1

Emri2 Mbiemri2, nr. i ID2

Emri3 Mbiemri3, nr. i ID3

1. Hyrje
   Përshkruaj detyrën.

2. Dizajni
   Përshkruaj se si janë të organizuar fajllat dhe çfarë module ka në secilin prej tyre. Çfarë pune kryejnë këto module.

3. Ekzekutimi
   Përshkruaj ku është fajlli i Testbench dhe si të ekzekutohet.

4. Përfundimi
   Në këtë pjesë përshkruani konkluzionet tuaja nga përvoja me këtë detyrë. Kjo pjesë nuk është e domosdoshme.

Raporti të mos jetë më i gjatë se 10 faqe.

7. Vlerësimi

Vleresimi behet ne forme te kombinuar: nje pjese te pikëve e merrni prej punimit te projektit dhe pjesën kryesore te pikëve e merrni nga prezantimi i projektit.

Pikët janë akumulative.

Ju keni mundësi për këto kombinime:

- Zgjedh vetëm pikën a [max. 5 pikë]; Raporti max. 2 pikë
- Zgjedh a, b, c [max. 11 pikë] (Vetëm a dhe b, apo a dhe c nuk lejohet – Numërohet si vetëm a); Raporti max. 3 pikë
- Zgjedh pikat a, b, c, d, e [max. 30 pikë]; Raporti max. 5 pikë.
- Pika g mund të kombinohet me cilëndo nga kombinimet e mësipërme për max. 10 pikë shtesë.

Piket e mesiperme përpjesëtohen me peshën e vlerësimit te punimit dhe jo me ate te prezentimit.