

ARKITEKTURA E KOMPJUTERËVE

Viti Akademik 2023/2024

DETYRA 2

Valon Raca & Synim Selimi

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

- Si pikë startuese e këtij projekti duhet të jetë dizajni i ALU-së 32 bitëshe që e kemi bërë në Javën e IV (Shtojca B.5 e librit kryesor, Ushtrimet në Verilog: Java VIII - 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.
- Si bazë për kontroll të përdoret Njësia e Kontrollit të cilën e kemi punuar në ushtrime.
- Register File, Data Memory dhe Instruction Memory do t'iu jepen të gatshme në javën e X-XII.

3.2. Specifikimi i sistemit

CPU duhet të jetë 16 bitëshe (half-word / 2 bajtësh).

Formati i instruksioneve

CPU duhet të përkrah instruksione të formatit R dhe I.

Formati R

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

Instruktionet e formatit R OPCODE do ta kenë 0XXX (XXX -> 000-111). Jo të gjitha instruksionet e formatit R do ta kenë OPCODE-in e njejte.

Formati I

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

Instrukcionet e formatit I OPCODE do ta kenë 1XXX (XXX -> 000-111). Instrukcionet e formatit I, secila vec e vec do te kete OPCODE te ndryshëm.

ALU

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

ALU do të ketë dy hyrje 16-biteshe A dhe B, dhe nje-biteshe hyrjen CarryIn; një dalje 16-biteshe për Rezultat, një dalje 1-biteshe për CarryOut, nje dalje 1-biteshe Zero, dhe nje dalje 1-biteshe per Overflow.

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

Njësia e kontrollit

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

Daljet e njësisë së kontrollit:

input	input.gjatësia[bit]	output	output.gjatësia [bit]
OPCODE	4	RegDst	1
		ALUSrc	1
		MemToReg	1
		RegWrite	1
		MemRead	1
		MemWrite	1
		ALUOp	2
		Branch	1

ALU Control Hyrjet

Në hyrje kemi 2 bit nga fusha FUNCT e instruksionit dhe 2 bit nga ALUOp.

ALUOp përcakton se për çfarë kryhet operacioni:

ALUOp	Selekto	Arsyeja
00	Mbledhje	LW apo SW
01	Zbritje	BEQ/BNE
10	Sipas FUNCT	Operacionet ALU
11	Ne baze te OPCODE	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	0	ADDI
1	1	0	0	SUB
1	1	0	0	SUBI
0	0	0	1	SLTI
0	1	1	0	SLL
0	1	1	1	SRA

Register File

Numri i regjistrave do të jetë 4 (regjistri \$zero dhe 3 të tjerë për përdorim të përgjithshëm). Regjistrat do të jenë të gjerë 16 bit.

Register File ka tre hyrje dy 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 RT.

Adresat e regjistrave:

\$zero	00
\$r1	01
\$r2	10
\$r3	11

Regjistri PC po ashtu është 16 bitësh dhe është i ndarë nga Register File. Për PC të krijohet mbledhës i veçantë. Inkrementuesi e rrit PC për 2 (2 bajt / 16 bit) në të gjitha rastet përveç kur kemi degëzime përmes BEQ/BNE.

Memoria

Dy memorie nga 128 bajt secila (Instruction dhe Data Memory) Harta e memories:

Adresa	Segmenti
0 – 9	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		r		RD		SHAMT				FUNCT	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	X	X	Y	Y	Z	Z	0	0	0	0	0	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 Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	0	0	1	0	0	10	0

OR

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

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 Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	0	0	1	0	0	10	0

XOR

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

Formati: R

Shablloni: XOR rd, rs, rt

Shembull: XOR \$r1, \$r2, \$r3

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

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	0	0	1	0	0	10	0

ADD

OPCODE				RS		I		RD		SHAMT				FUNCT	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	X	X	Y	Y	Z	Z	0	0	0	0	0	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 Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	0	0	1	0	0	10	0

ADDI

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

Formati: I

Shablloni: ADDI rt, rs, immediate

Shembull: ADDI \$r1, \$r2, 5

Kryhet mbledhje ndërmjet vlerës së regjistrit \$r2 dhe vlerës imediate 5 (shembull). Rezultati ruhet në \$r1.

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
0	1	0	1	0	0	11	0

SUB

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

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 Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	0	0	1	0	0	10	0

SUBI

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

Formati: I

Shablloni: SUBI rt, rs, immediate

Shembull: SUBI \$r1, \$r2, 5

Kryhet zbritje ndërmjet vlerës së regjistrit \$r2 dhe vlerës imediate 5 (shembull). Rezultati ruhet në \$r1.

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
0	1	0	1	0	0	11	0

SLTI

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

Formati: I

Shablloni: SLTI rt, rs, immediate

Shembull: SLTI \$r1, \$r2, 5

Kryhet funksioni SET LESS THAN ne mes regjistrit \$r2 dhe vlerës imediate. Biti i fundit (LSB) behet 1 nese \$r2 < 5 (bitat tjerë 0), dhe ne te kundërtën LSB behet 0.

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
0	1	0	1	0	0	11	0

INSTRUKSIONET MEMORIKE LW

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

1	1	0	0	X	X	Y	Y	Z	Z	Z	Z	Z	Z	Z	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Formati: I

Shablloni: LW rt, (immediate)rs

Shembull: LW \$r1, 8(\$r2)

Vlera imediate e instruksionit 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 memorien e të dhënave. Vlera që lexohet nga ajo adresë ruhet në \$r1.

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
0	1	1	1	1	0	00	0

SW

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

Formati: I

Shablloni: SW rt, (immediate)rs,

Shembull: SW \$r1, 2(\$r2)

Vlera imediate e instruksionit 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 memorien e të dhënave. Vlera që gjendet në \$r1 shkruhet në adresën e përcaktuar në memorie.

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
0	1	0	0	0	1	00	0

INSTRUKSIONET E DEGEZIMIT BEQ

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

1	1	1	1	X	X	Y	Y	Z	Z	Z	Z	Z	Z	Z	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Formati: I

Shablloni: BEQ rs, rt, L1

Shembull: BEQ \$r1, \$r2, kercimi

Vlera imediate e instruksionit zgjerohet nga 8-bit në 16-bit. Vlera e zgjeruar shtyhet për një bit majtas (shumëzim me 2, pasi instruksionet 2 bajtëshe).

Zbritet \$r2 nga \$r1, nëse rezultati nuk është zero biti Zero i ALU bëhet 0, që sinjalizon se vlera e PC+2 duhet të mblidhet me vlerën imediate të përpunuar më lartë dhe të vendoset në PC.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
0	0	0	0	0	0	01	1

INSTRUKSIONET E SHTYERJES (BONUS) SLL

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

Formati: R

Shablloni: SLL rd, rs, SHAMT

Shembull: SLL \$r1, \$r2, 9

Kryhet zhvendosje logjike majtas e bitave të \$r2 për aq bita sa parashihet në SHAMT. Rezultati ruhet në \$r1.

Vlerat dalëse të Njësisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	X	0	1	0	0	10	0

SRA

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

Formati: R

Shablloni: SRA rd, rs, SHAMT

Shembull: SRA \$r1, \$r2, 9

Kryhet zhvendosje aritmetike djathtas e bitave të \$r2 për aq bita sa parashihet në SHAMT.

Rezultati ruhet në \$r1. Ruhet shenja e numrit.

Vlerat dalëse të Njesisë Kontrolluese:

RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	ALUOp	Branch
1	X	0	1	0	0	10	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
- 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)
- Të shtohet ALU Control sipas dizajnit nga ligjëratat
 - Të ndryshohet Njësia e Kontrollit dhe ALU Control varësisht nga instruksionet që shtohen
- d) [7 pikë] Instruksionet bazë që duhet të përkrahen nga CPU
- and, or, xor (1 pikë) ii. add, addi, sub, subi (1 pikë)
 - lwi (1 pikë) iv. sw (1 pikë)
 - beq (3 pikë)

- e) [7 pikë] Të ekzekutohet programi i mëposhtëm nga memoria:

```
sub $r1, $zero, $zero
lw $r2, 4($r1)
xor $r1, $r2, $r3
bne $r2, $zero, kercimi
and $r3, $r1, $r2
kercimi: sw $r3, 4($r2)
slti $r2, $r3, 2 #vetëm nëse është implementuar instruksioni nga bonus
sll $r1, $r2, 3 #vetëm nëse është implementuar instruksioni nga bonus
sra $r2, $r1, 9 #vetëm nëse është implementuar instruksioni nga bonus
```

- f) [5 pikë] Raporti përfundimtar *Opsionale [Bonus pikë – max. 10 pikë]*
- g) [0 - 10 pikë] Të shtohen instruksionet e mëposhtme
 - i. [2 pikë] SLTI ii. [4 pikë] SLL
 - iii. [4 pikë] SRA

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