

VISOKA TEHNIČKA ŠKOLA  
STRUKOVNIH STUDIJA  
SUBOTICA



ARITMETIČKI MODUL

Projekat iz predmeta:

Arhitektura računara

Mentor: dr. Tibor Sakal  
profesor strukovnih studija

Kandidat: Stepan Turicin  
br. indeksa: 26123036  
Studijski program: Informatika

# Sadržaj

Sadržaj .....	2
Uvod .....	3
1. Projektni zadatak .....	4
1.1 Svrha projekta i ograničenja .....	4
1.2 Način realizacije.....	4
2. Teorijske osnove.....	5
2.1 Aritmetički deo .....	5
2.1.1 Modul za sabiranje i oduzimanje .....	5
2.1.2 Modul za unaprednu inverziju.....	10
2.1.3 Modul za inkrement i dekrement broja A .....	11
2.2 Logički deo .....	12
2.2.1 Logičko I (AND).....	12
2.2.2 Logičko ILI (OR).....	12
2.2.3 Logičko NE (NOT).....	12
2.2.4 Isključivo ILI (XOR).....	13
2.3 Dodatni moduli .....	13
2.3.1 Overflow modul .....	13
2.3.2 Buffer modul.....	14
2.3.3 Dekoder sa negativnom logikom.....	15
2.4 Unutar aritmetičkog modula .....	16
3. Prikaz simulacije.....	17
4. Prilog.....	20
5. Literatura.....	26

# **Uvod**

Ovaj dokument pruža detaljan opis funkcionisanja i izrade zadatka na temu “Aritmetički modul”.

U dokumentu će biti predstavljeni sledeći elementi:

- Jednačine funkcionisanja korišćenih elemenata: Ovo uključuje matematičke modele koji opisuju ponašanje i karakteristike elektronskih komponenti koje se koriste u projektu.
- Tablice istine: Biće prikazane tablice koje ilustruju logičke operacije i ponašanje kola u različitim uslovima.
- Šeme i slike: Vizuelni prikazi sklopova i kola napravljeni u programu Proteus, koji će pomoći u boljem razumevanju koncepta i dizajna projekta.

Seminarski rad uključuje dokumentaciju u formatu .doc, i projekat u .pdspj formatu, koji je kreiran u Proteus-u. Ovaj projekat omogućava simulaciju rada sklopova, što je ključno za verifikaciju pravilnosti rada kola pre nego što se realizuje fizička implementacija.

# **1. Projektni zadatak**

Izabrana tema: 36-E4

Naziv teme: Aritmetički modul. A i B su 16 bitni brojevi sa predznakom. A+B, A-B, -A, -B, A+1, A-1

Dodatni zahtev: Bez namenskih kola

## **1.1 Svrha projekta i ograničenja**

Cilj ovog projekta je da prikaže funkcionisanje aritmetičkog modula kroz simulaciju u programu Proteus, bez korišćenja namenskih kola. Dodatni cilj je demonstrirati da je moguće implementirati svako namensko kolo koristeći osnovne elektronske komponente. Ograničenje projekta je rad sa 16-bitnim brojevima sa predznakom, gde su brojevi A i B u opsegu od -32,768 do 32,767. Simulacija obuhvata sledeće operacije: A+B, A-B, -A, -B, A+1 i A-1, koje će biti testirane za sve moguće kombinacije dva 16-bitna broja.

## **1.2 Način realizacije**

Projekat je realizovan putem kreiranja osnovnih modula, poput 1-bitnog sabirača, iz kojih su kasnije sastavljeni složeniji komponente. Na primer, 16-bitni sabirač i oduzimač, namenjeni za izvođenje operacija sabiranja i oduzimanja dva broja A i B. Ove osnovne aritmetičke operacije postale su osnova za implementaciju drugih operacija, kao što su -A, -B, A+1 i A-1, pomoću 16-bitnog sabirača i oduzimača.

U finalnoj implementaciji sve aritmetičke operacije se biraju pomoću 3x8 dekodera, što omogućava fleksibilno upravljanje izborom potrebne operacije. Finalna šema projekta biće prikazana u obliku električne šeme, koja prikazuje rad svih komponenti tokom izvršavanja operacija.

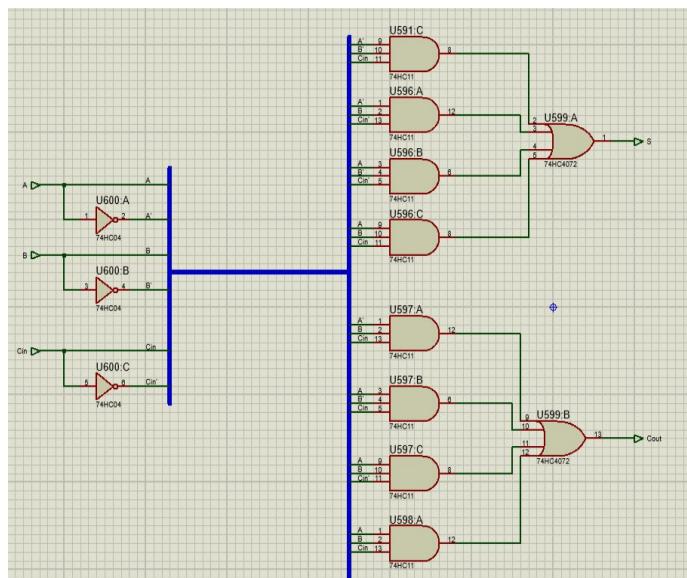
## 2. Teorijske osnove

### 2.1 Aritmetički deo

#### 2.1.1 Modul za sabiranje i oduzimanje

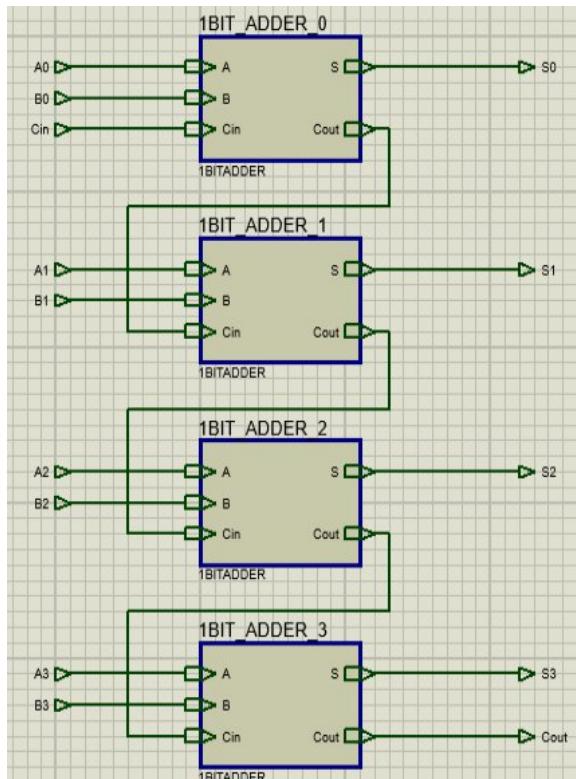
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabela 1. – tablica istinitosti za 1-bitni pun sabirač

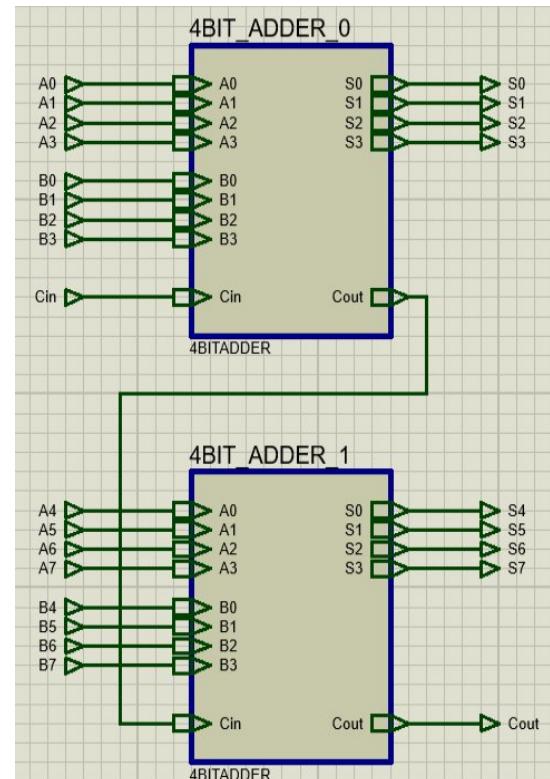


Slika 1.- šema 1-bitnog punog sabirača

Koristeći AND i NOT logičke operacije, kreiramo 1-bitni sabirač koji prima ulaze A, B i Carry In, a proizvodi izlaze S (zbir) i Carry Out (prenos).



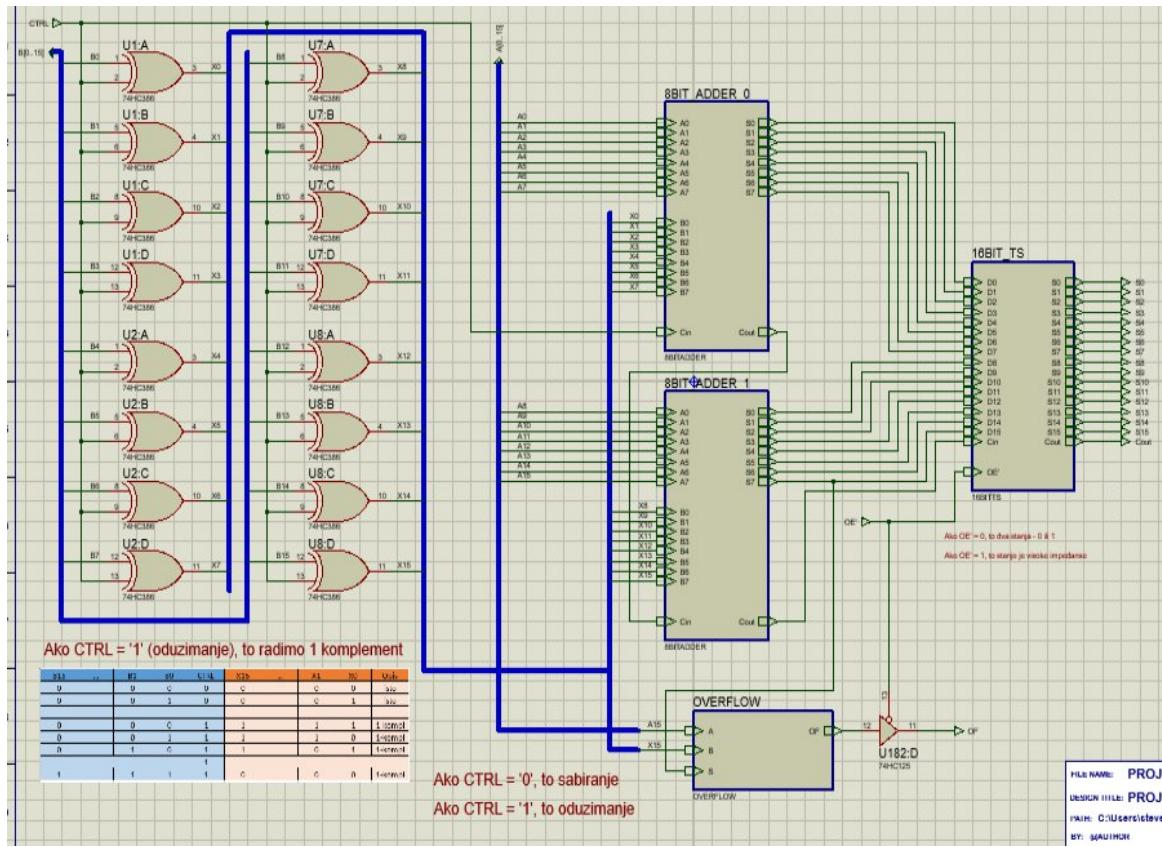
Slika 2.- šema 4-bitnog punog sabirača



Slika 3.- šema 8-bitnog punog sabirača

Povezivanjem 1-bitnih sabirača u lanac, gde se Carry Out iz svakog prethodnog sabirača koristi kao Carry In za sledeći, napravljen je 4-bitni sabirač (iz slike 2.). Ovaj postupak omogućava sabiranje četiri bita brojeva A i B uz pravilan prenos između bitova.

Na isti način, povezujući dva 4-bitna sabirača, kreiran je 8-bitni sabirač (iz slike 3.).



Slika 4.- šema 16-bitnog sabirača i oduzimača

Nakon toga je napravljen modul "Sabirača i Oduzimača" (iz slike 4.) omogućava izvođenje operacija sabiranja i oduzimanja, pri čemu se rezultat generiše u obliku 16-bitnog broja S (S0..S15), uz dodatne signale Carry i Overflow (OF) za detekciju prenosa i prekoračenja opsega.

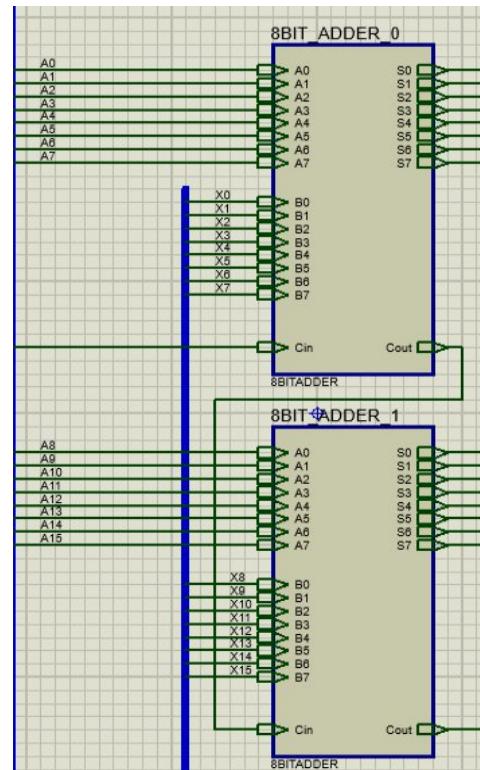
Ulazi:

- Bus za 16-bit brojeve A i B (A0..A15 i B0..B15)
- Kontrolni signal CTRL: Mogućnost izbora operacije (0 za sabiranje, 1 za oduzimanje)
- Kontrolni signal OE': Omogućava isključenje izlaza modula pomoću Tri-State buffer kada modul nije u upotrebi (1 je uključen, 0 je isključen).

U nastavku razložimo konkretnije svaki deo ovog modula.



Slika 5.- šema prvog komplementa B broja ili samo B



Slika 6.- šema 16-bitnog punog sabirača

U ovom delu (iz slike 5.) modula koristi se logička operacija XOR između signala CTRL i svakog bita broja B.

- Ako je CTRL jednak 1 (oduzimanje), svaki bit broja B se inverte, što odgovara operaciji dobijanja prvog komplementa.
- Ako je CTRL jednak 0 (sabiranje), bitovi broja B ostaju nepromjenjeni i prenose se u istom obliku.

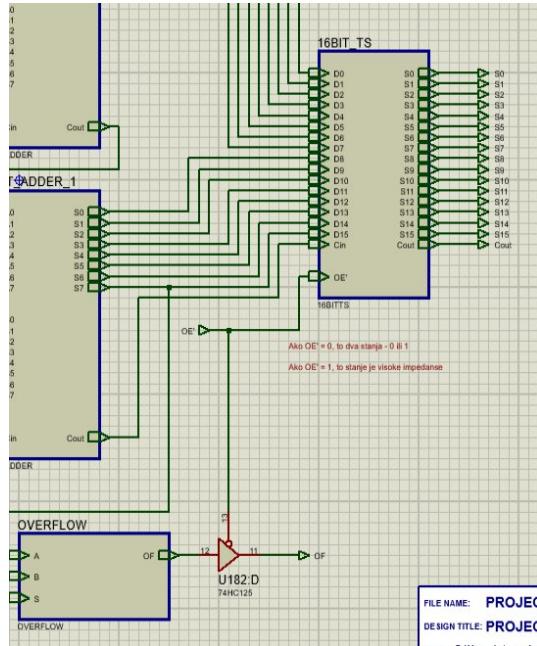
B15	...	B1	B0	CTRL	X15	...	X1	X0	Opis
0		0	0	0	0		0	0	isto
0		0	1	0	0		0	1	isto
0		0	0	1	1		1	1	1-kompl
0		0	1	1	1		1	0	1-kompl
0		1	0	1	1		0	1	1-kompl
					1				
1		1	1	1	0		0	0	1-kompl

Tabela 2. – tablica istinitosti za 1-bitni pun sabirač

Dobijeni bitovi se označavaju kao X0..X15 i prosleđuju se u magistralu za dalje izvođenje operacija sabiranja ili oduzimanja.

Nakon toga, brojevi se sabiraju u dva međusobno povezana 8-bitna sabirača (iz slike 6.).

- U ove sabirače se prosleđuju ulazni brojevi A0..A15, X0..X15, kao i signal CTRL na ulaz Carry In.
- Ako je u pitanju operacija oduzimanja, u Carry In prvog 8-bitnog sabirača šalje se jedinica, čime se ostvaruje drugi komplement broja.



Slika 7.- izlaze za modul “Sabirač i oduzimač”

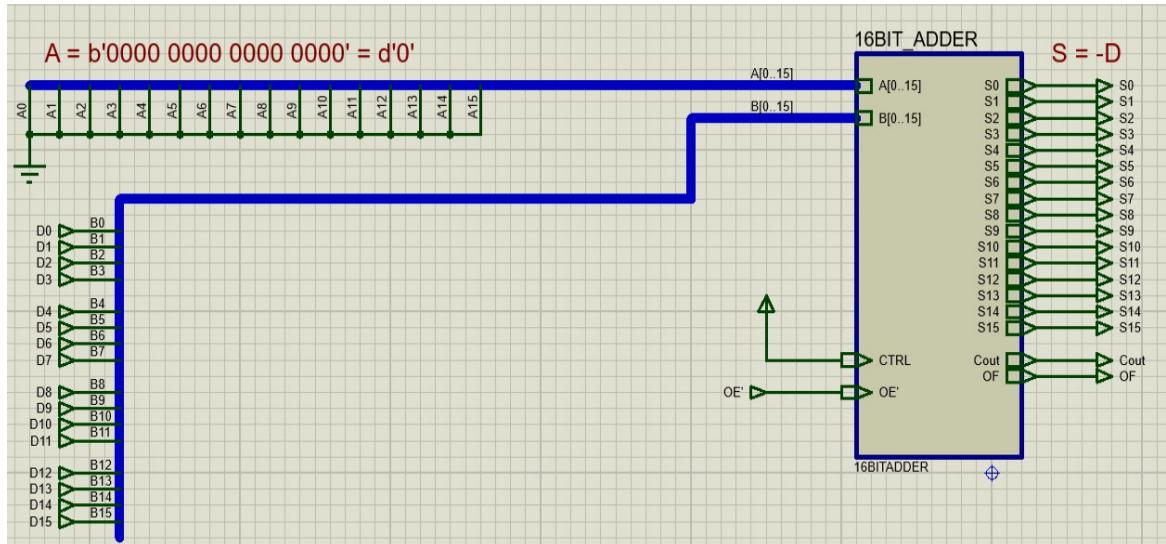
Posle toga koristi se modul Overflow (iz slike 7.) za poslednje bitove X15, A15 i S15 kako bi se proverilo da li je došlo do prekoračenja (Overflow).

Na kraju, svi izlazi prolaze kroz modul 16BIT\_TS (Tri-State Buffer) (iz slike 7.), koji omogućava uključivanje ili isključivanje izlaza prema potrebi.

### 2.1.2 Modul za unaprednu inverziju

D15	...	D1	D0	OE'	S15	...	S1	S0	Cout (Carry)	OF
x		x	x	1	hi-Z		hi-Z	hi-Z	hi-Z	hi-Z
0		0	0	0	0		0	0	1	0
0		0	1	0	1		1	1	0	0
0		1	0	0	1		1	0	0	0
0		1	1	0	1		0	1	0	0
				0					...	
1		1	0	0	0		1	0	0	0
1		1	1	0	0		0	1	0	0

Tabela 3. – tablica istinitosti za 16-bitni inverziju



Slika 8.- šema za modul inverziji

Ovaj modul prima 16-bitna broj (od D0 do D15) i signal OE'. Modul funkcioniše prema principu  $0 - D = \text{inverzija}$ , gde se broj D predstavlja na magistrali kao B0..B15, dok je broj A postavljen na nulu.

Zatim, oba broja se prosleđuju u "16-bitni Sabirač i Oduzimač", koji izvršava operaciju oduzimanja. Na izlazu dobijamo rezultat ove operacije.

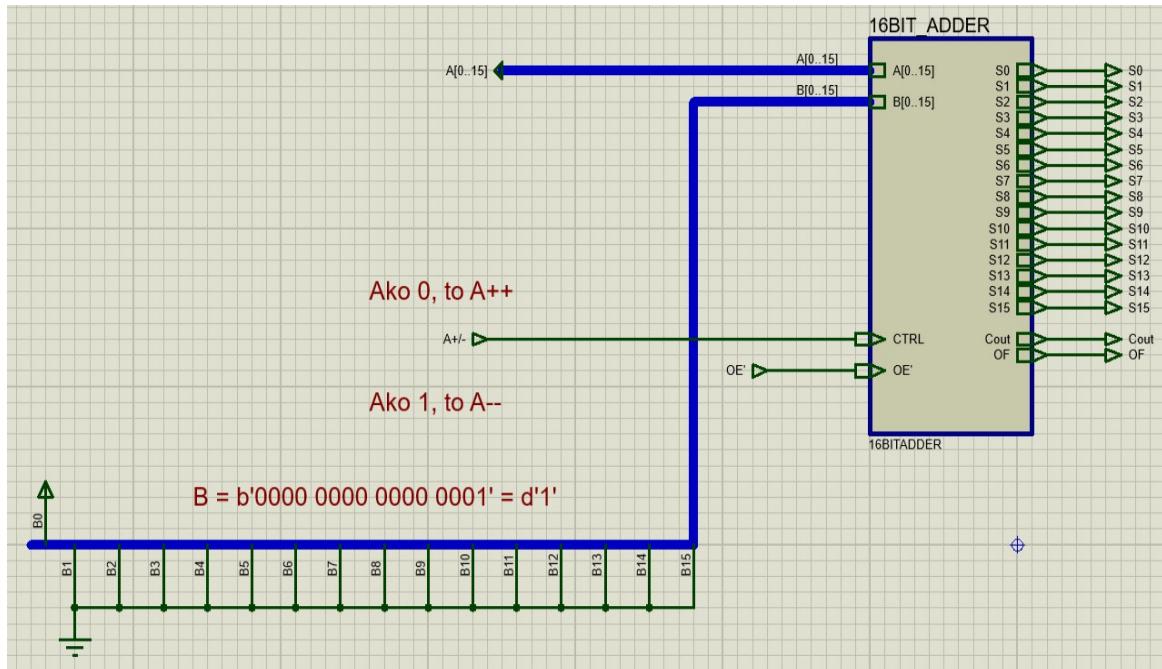
### 2.1.3 Modul za inkrement i dekrement broja A

A15	...	A1	A0	A+/-	OE'	S15	...	S1	S0	Cout (Carry)	OF
x		x	x	x	1	hi-Z		hi-Z	hi-Z	hi-Z	hi-Z
0		0	0	0	0	0		0	1	0	0
0		0	1	0	0	0		1	0	0	0
0		1	0	0	0	0		1	1	0	0
	...				0	0					
1		1	0	0	0	1		1	1	0	0
1		1	1	0	0	0		0	0	1	0

Tabela 4. – tablica istinitosti za inkrement (za  $A+/-$  je 0)

A15	...	A1	A0	A+/-	OE'	S15	...	S1	S0	Cout (Carry)	OF
x		x	x	x	1	hi-Z		hi-Z	hi-Z	hi-Z	hi-Z
0		0	0	1	0	1		1	1	0	0
0		0	1	1	0	0		0	0	1	0
0		1	0	1	0	0		0	1	1	0
	...				1	0					
1		1	0	1	0	1		0	1	1	0
1		1	1	1	0	1		1	0	1	0

Tabela 5. – tablica istinitosti za dekrement (za  $A+/-$  je 1)



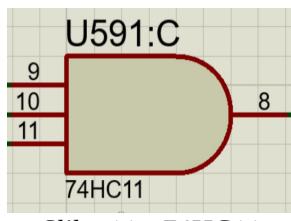
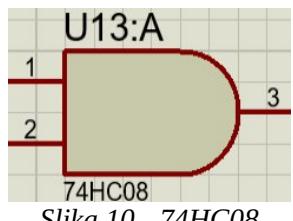
Slika 9.- Šema za modul inkrement/dekrement

U ovom modulu koristi se "16-bitni Sabirač i Oduzimač". Na ulaz A se prosleđuje 16-bitni broj, dok se na ulaz B šalje jedinica (0000 0000 0000 0001). Zatim se izvršava operacija sabiranja ili oduzimanja, zavisno od signala CTRL.

## 2.2 Logički deo

### 2.2.1 Logičko I (AND)

Za logičko I koristili su se elementi kao što su 74HC08 sa 2 ulaza i 74HC11 sa tri.



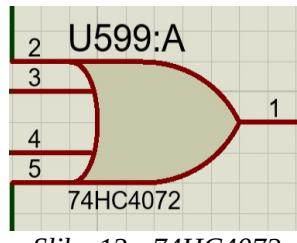
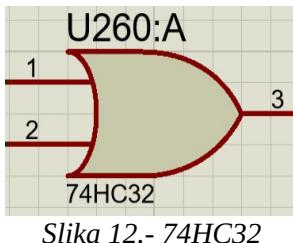
A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 6. – tablica istinitosti za AND

Funkcija logičkog I (AND) može se opisati na sledeći način: "Ako su ulaze jednake 1, izlaz R će biti 1". Ovo možemo videti posmatranjem tabele istine.

### 2.2.2 Logičko ILI (OR)

Za logičko ILI koristili su se elementi kao što su 74HC32 sa 2 ulaza i 74HC4072 sa četiri.



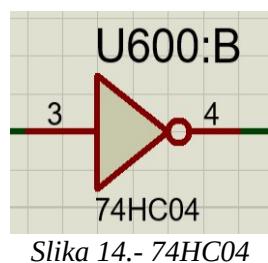
A	B	R
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 7. – tablica istinitosti za OR

Funkciju logičkog ILI kola možemo opisati na sledeći način: "Ako je A jednako 1 ILI je B jednako 1 (ili su oba ulaza jedinica), onda na izlazu R imamo jedinicu".

### 2.2.3 Logičko NE (NOT)

Logičko NE kolo ili inverzor je najjednostavnije logičko kolo (element 74HC04). Na izlazu daje negiranu vrednost ulazne veličine.

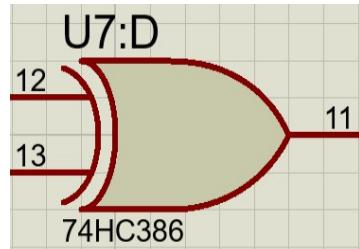


A	R
0	1
1	0

Tabela 8. – tablica istinitosti za NOT

## 2.2.4 Isključivo ILI (XOR)

Funkcija XOR kola može se opisati sa: "Ako je bilo koji ulaz A ili B jedinica, ali NE oba, izlaz R je jedinica". Koristio sam element 74HC386.



Slika 15.- 74HC386

A	B	R
0	0	0
0	1	1
1	0	1
1	1	0

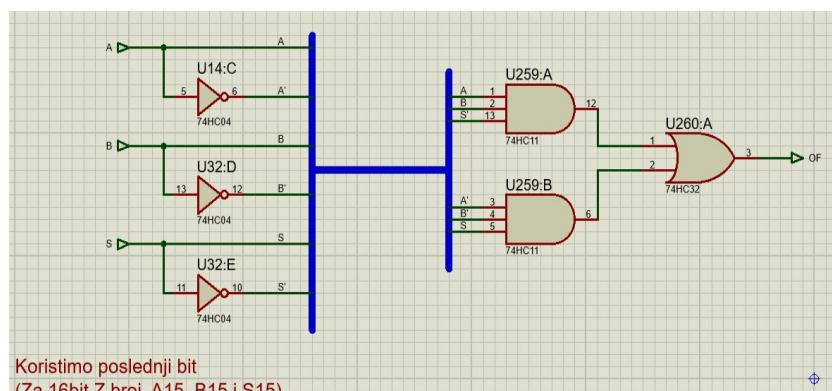
Tabela 9. – tablica istinitosti za XOR

## 2.3 Dodatni moduli

### 2.3.1 Overflow modul

A	B	S	OF
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Tabela 10.- tablica istinitosti za Overflow



Slika 16. – šema za overflow modul

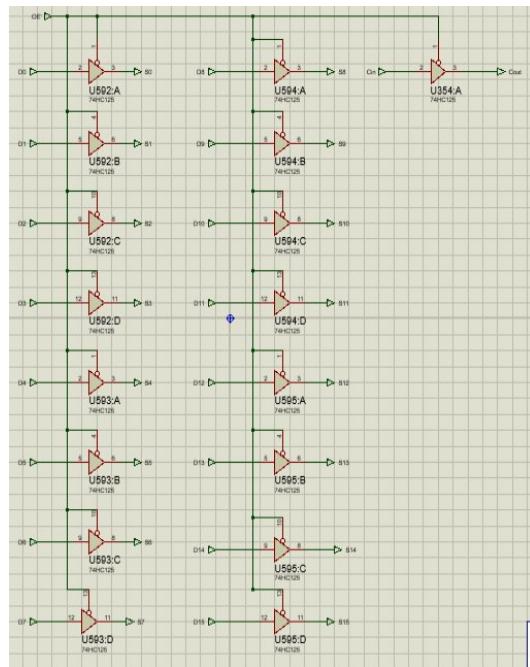
Upotreboom NOT, AND i OR kola proveravamo validnost izlaza prilikom sabiranja i oduzimanja sa predznakom. Overflow bit se postavlja na 1 ako su predznaci (MSB) ulaznih brojeva isti, ali predznak izlaza se ne poklapa sa njima. Ovo ukazuje na to da rezultat izlazi iz opsega koji može biti predstavljen sa 16 bitova.

Kada nisu aktivne aritmetičke operacije ili kada se koristi drugi komplement, overflow bit se postavlja u stanje visoke impedanse.

### 2.3.2 Buffer modul

D15	...	D1	D0	OE'	S15	...	S1	S0
x		x	x	1	hi-Z		hi-Z	hi-Z
0		0	0	0	0		0	0
0		0	1	0	0		0	1
...					...			

Tabela 11.- tablica istinosti za 16-bit Buffer



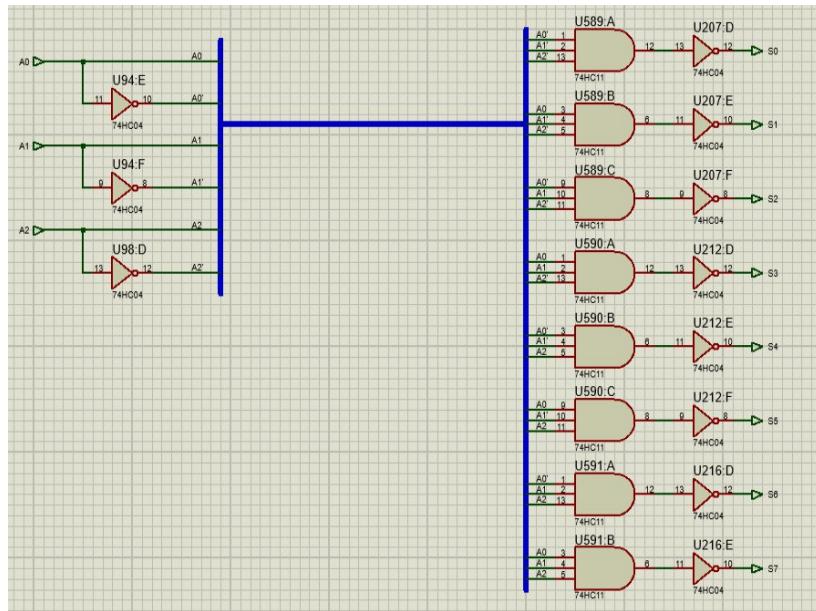
Slika 17. – šema za 16-bit buffer modul

Za 16-bitni sabirač i oduzimač napravljen je poseban modul sa Tri State Buffer-om za uključivanje i isključivanje rada modula. Kada je signal OE' postavljen na 1, izlaz modula prelazi u stanje visoke impedanse, što znači da modul ne utiče na magistralu. Kada je OE' = 0, modul je aktivran.

### 2.3.3 Dekoder sa negativnom logikom

A2	A1	A0	S7	S6	S5	S4	S3	S2	S1	S0	Operacija
0	0	0	1	1	1	1	1	1	1	0	A+B
0	0	1	1	1	1	1	1	1	0	1	A-B
0	1	0	1	1	1	1	1	0	1	1	-A
0	1	1	1	1	1	1	0	1	1	1	-B
1	0	0	1	1	1	0	1	1	1	1	A++
1	0	1	1	1	0	1	1	1	1	1	A--
1	1	0	1	0	1	1	1	1	1	1	None
1	1	1	0	1	1	1	1	1	1	1	None

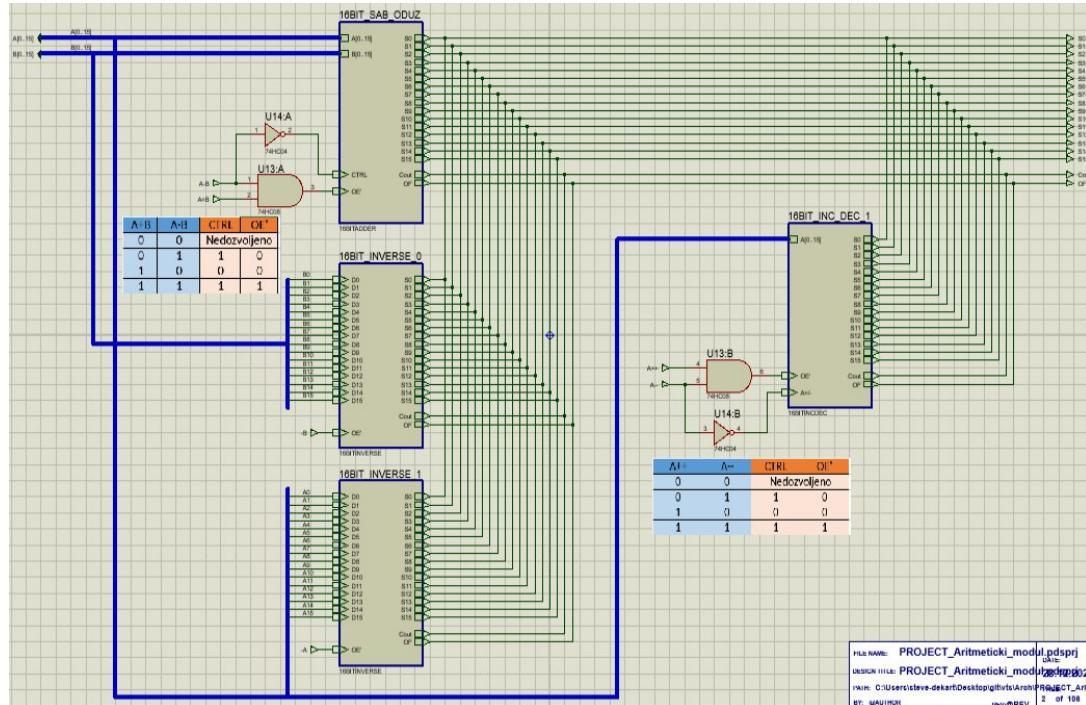
Tabela 12.- tablica istinosti za dekoder sa negativnom logikom



Slika 18. – šema za dekoder sa negativnom logikom

Za izbor konkretnе operacije razvijen je 3x8 dekoder, koji na osnovу ulaza od 3 bita bira određenu operaciju. Izabranoj operaciji biće dodeljena vrednost 0, dok će svim ostalim operacijama biti dodeljena vrednost 1.

## 2.4 Unutar aritmetičkog modula



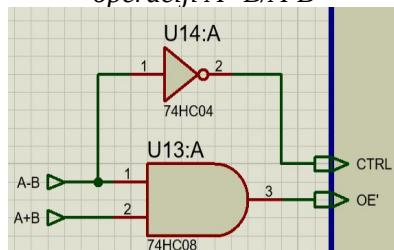
Slika 19. – šema za aritmetički modul

Unutar aritmetičkog modula (iz slike 18.) nalaze se četiri modula: 16-bitni sabirač i oduzimač, inkrement/dekrement za A, inverzija za A i inverzija za B. Svaki od njih je povezan sa svojom magistralom A i B, kao i ulazom koji zavisi od izabrane operacije putem OE'. Svi moduli imaju isti izlaz S0..S15, koji se bira pomoću OE'.

Za module "sabirač i oduzimač" za A i B, kao i "inkrement/dekrement" za A, kreirana je logička šema zasnovana na dve operacije unutar jednog modula. U nastavku su prikazane šeme i tabele koje objašnjavaju funkcionisanje izbora operacija.

A+B	A-B	CTRL	OE'
0	0	Nedozvoljeno	
0	1	1	0
1	0	0	0
1	1	1	1

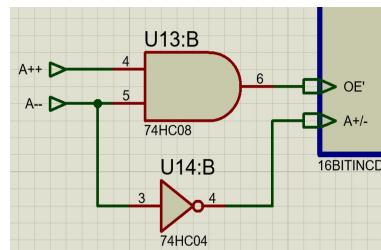
Tabela 13.- tablica istinosti za izbor operacija A+B/A-B



Slika 20. – šema za izbor operacija A+B/A-B

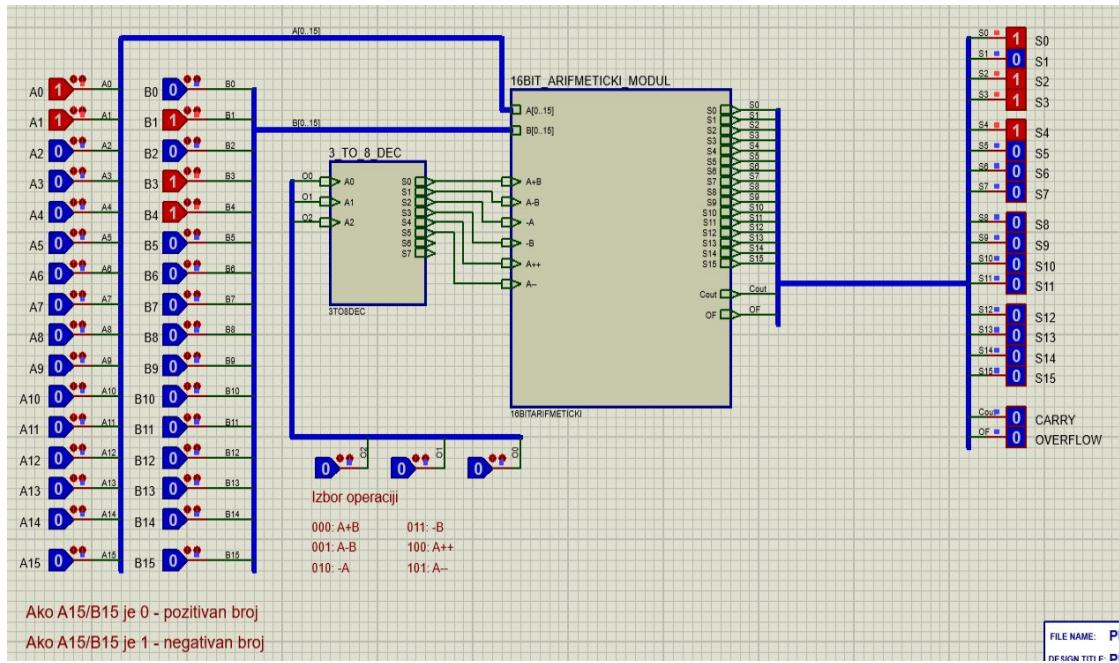
A++	A--	CTRL	OE'
0	0	Nedozvoljeno	
0	1	1	0
1	0	0	0
1	1	1	1

Tabela 14.- tablica istinosti za izbor operacija A++/A--

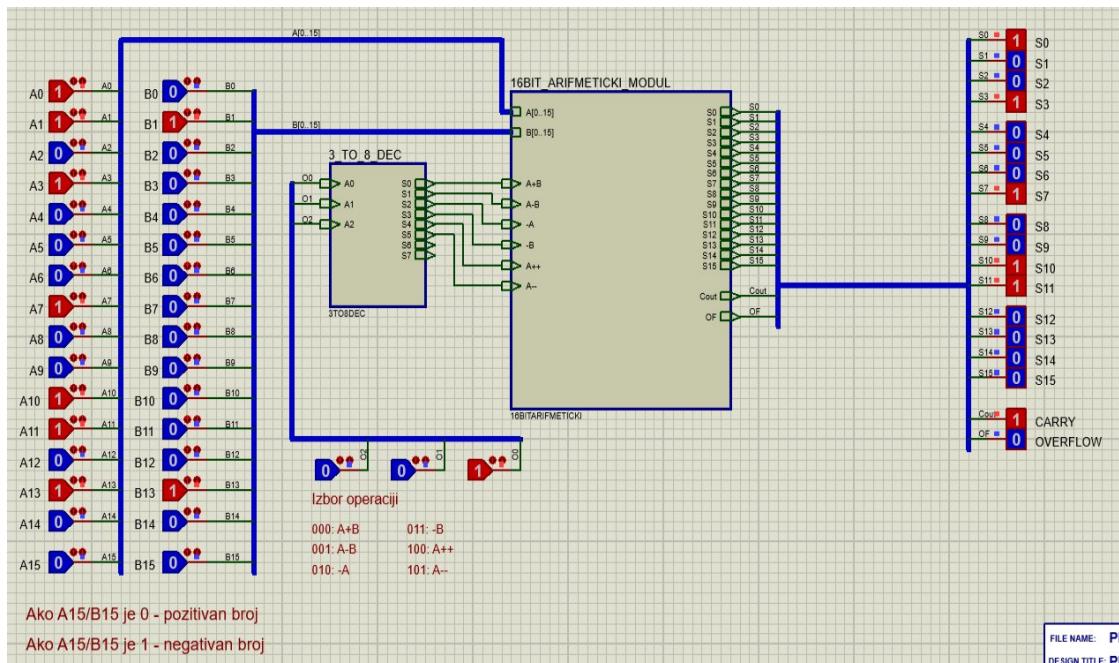


Slika 21. – šema za izbor operacija A++ ili A--

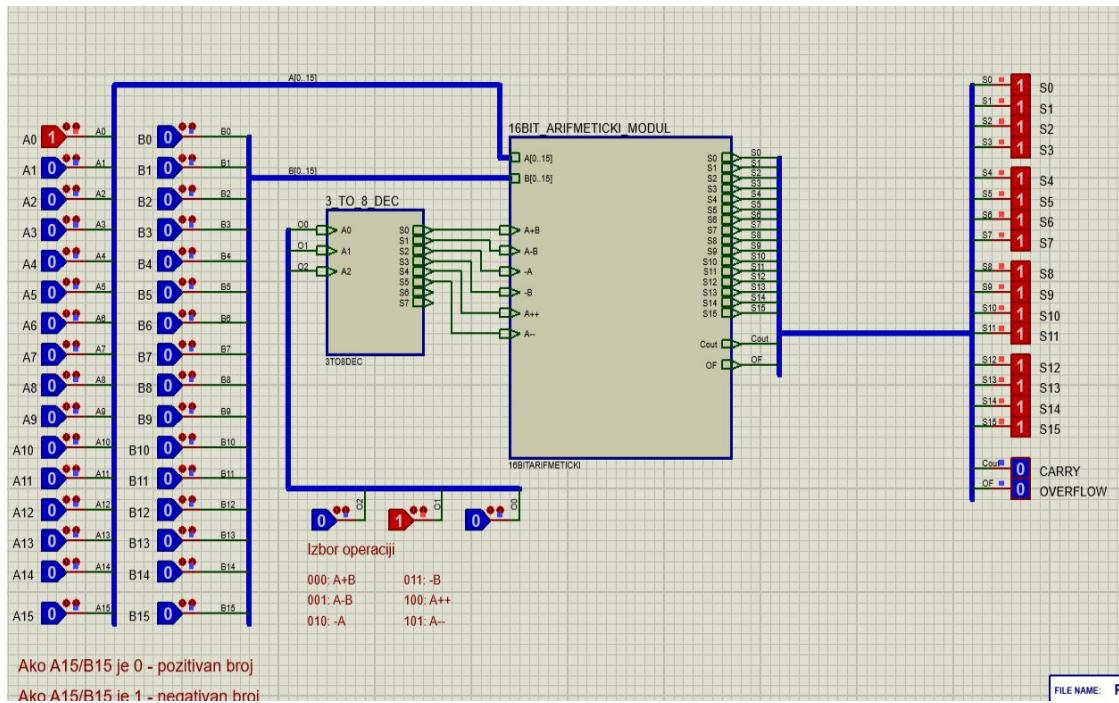
### 3. Prikaz simulacije



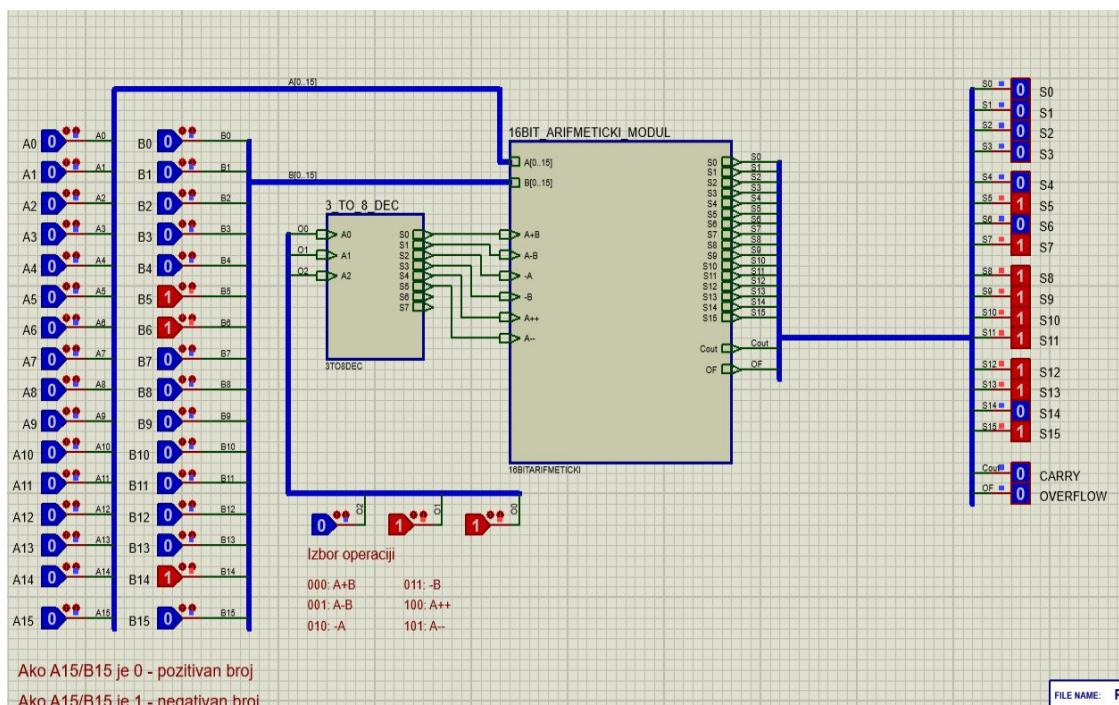
Slika 22. – Operacija A+B



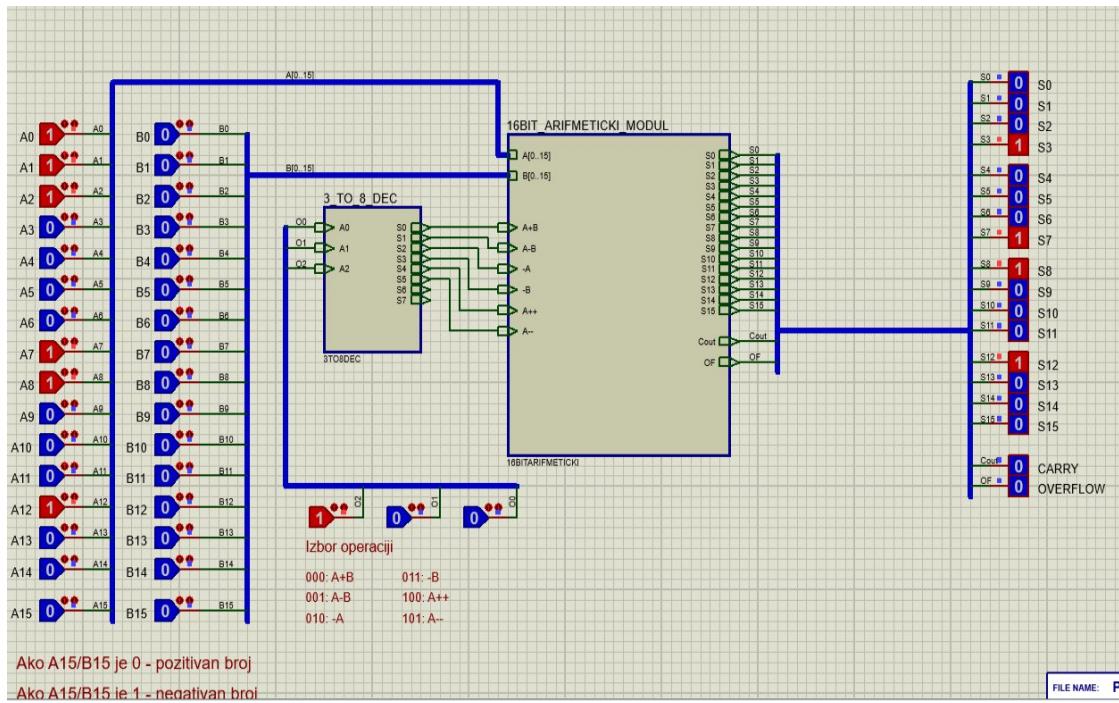
Slika 23. – Operacija A-B



Slika 24. – Operacija -A

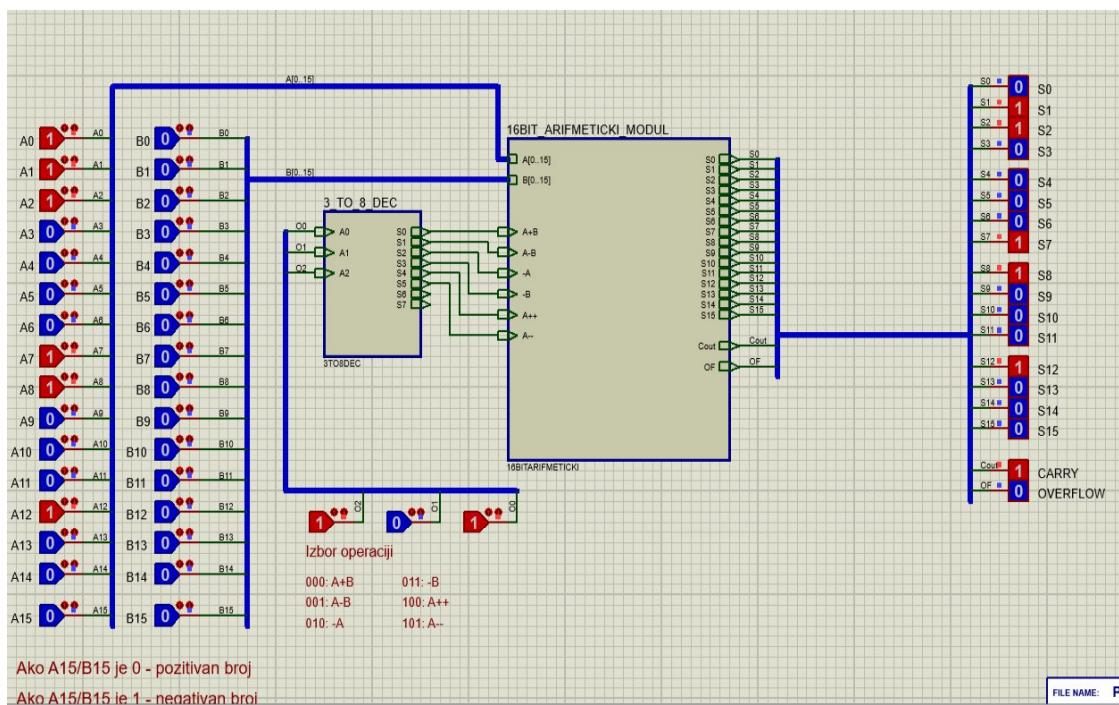


Slika 25. – Operacija -B



Slika 26. – Operacija  $A++$

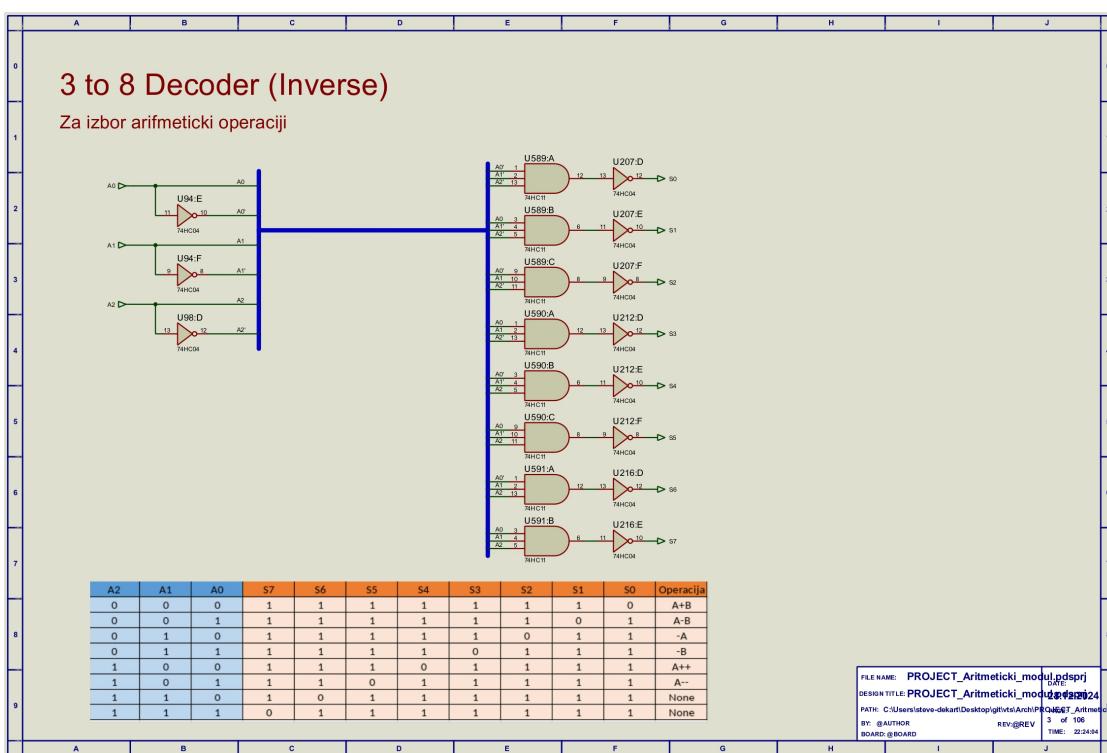
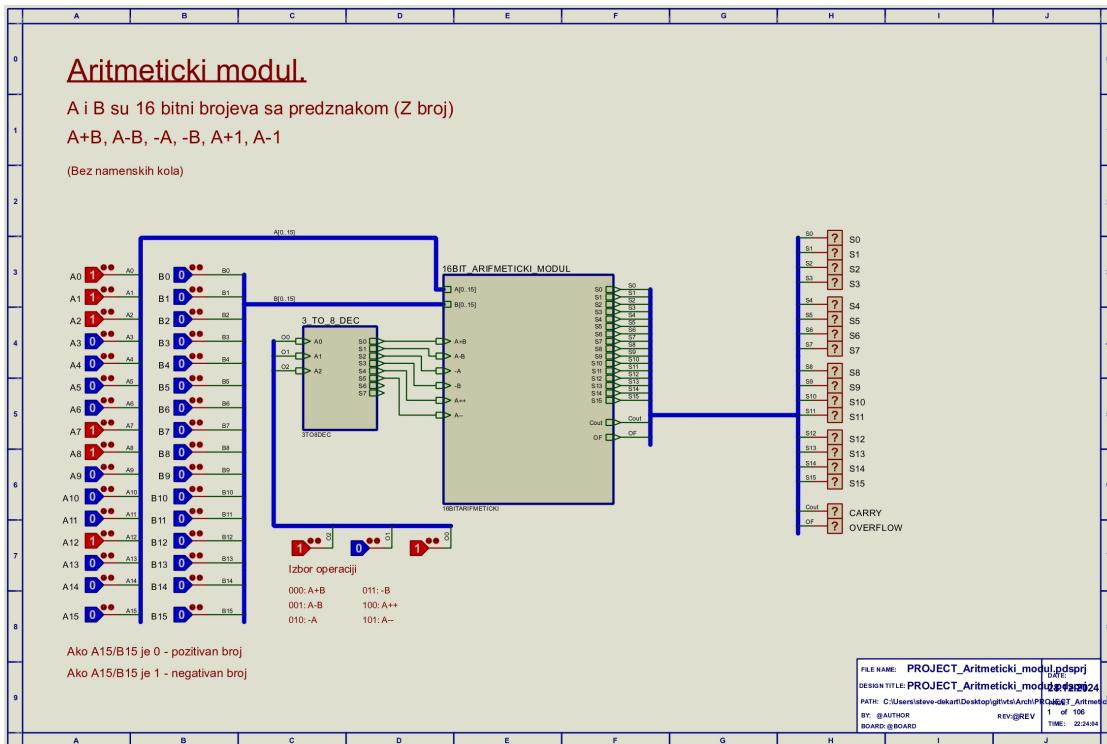
FILE NAME: P

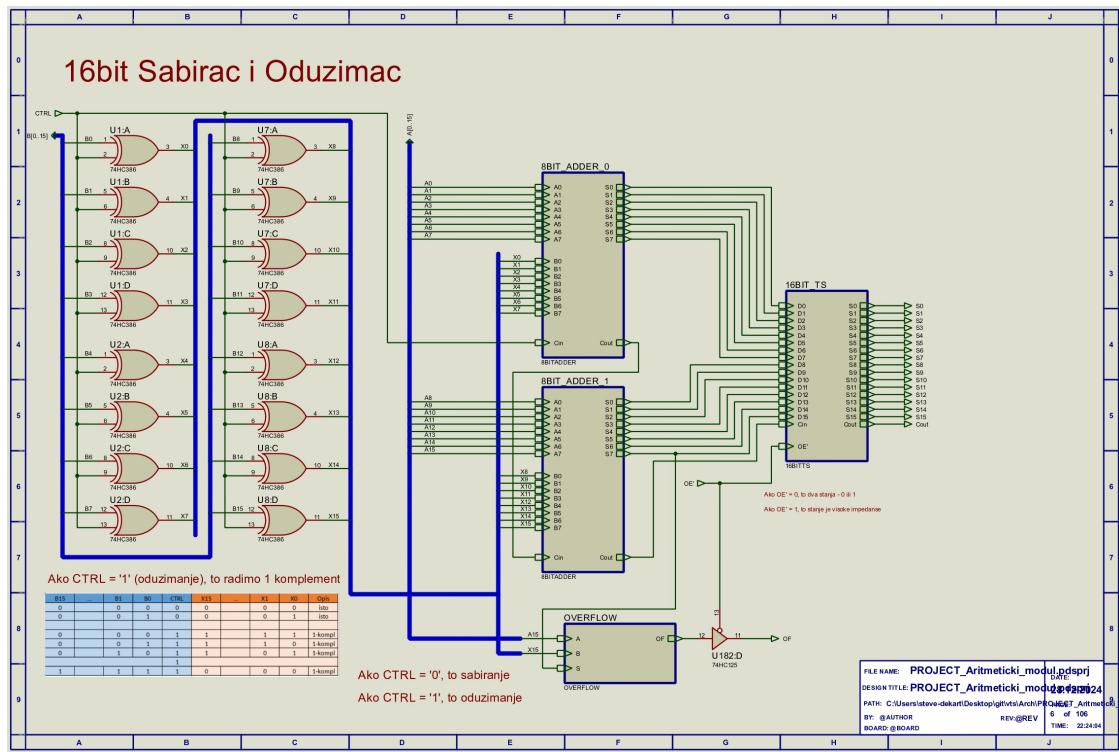
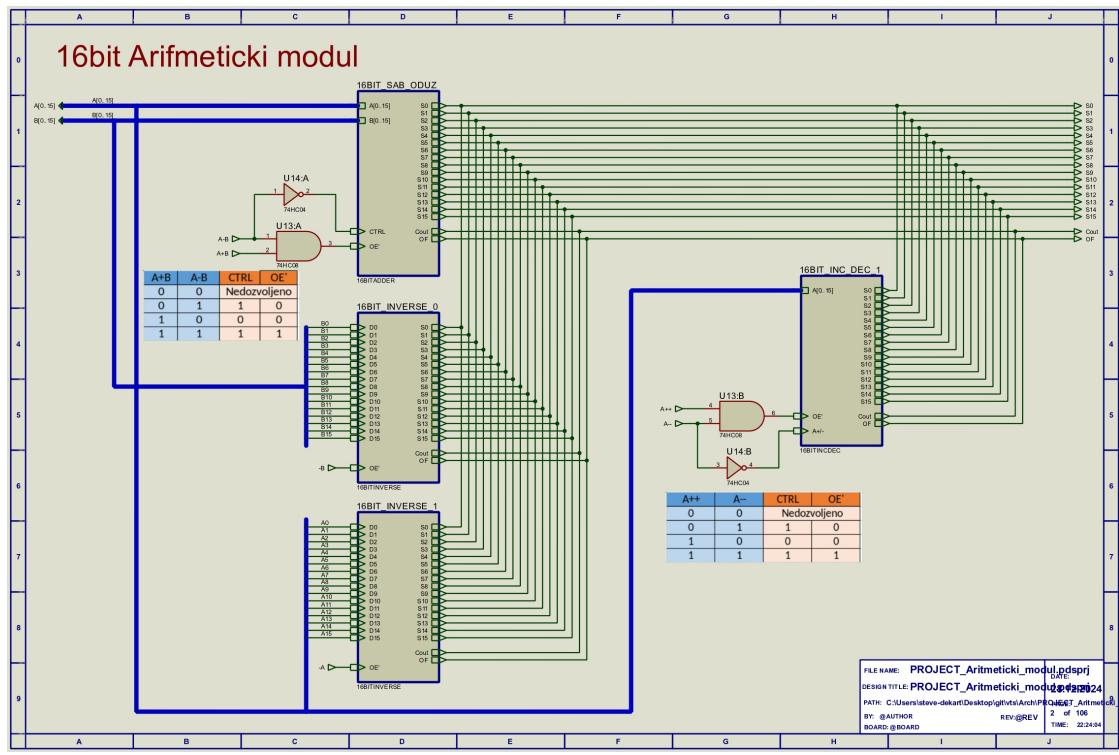


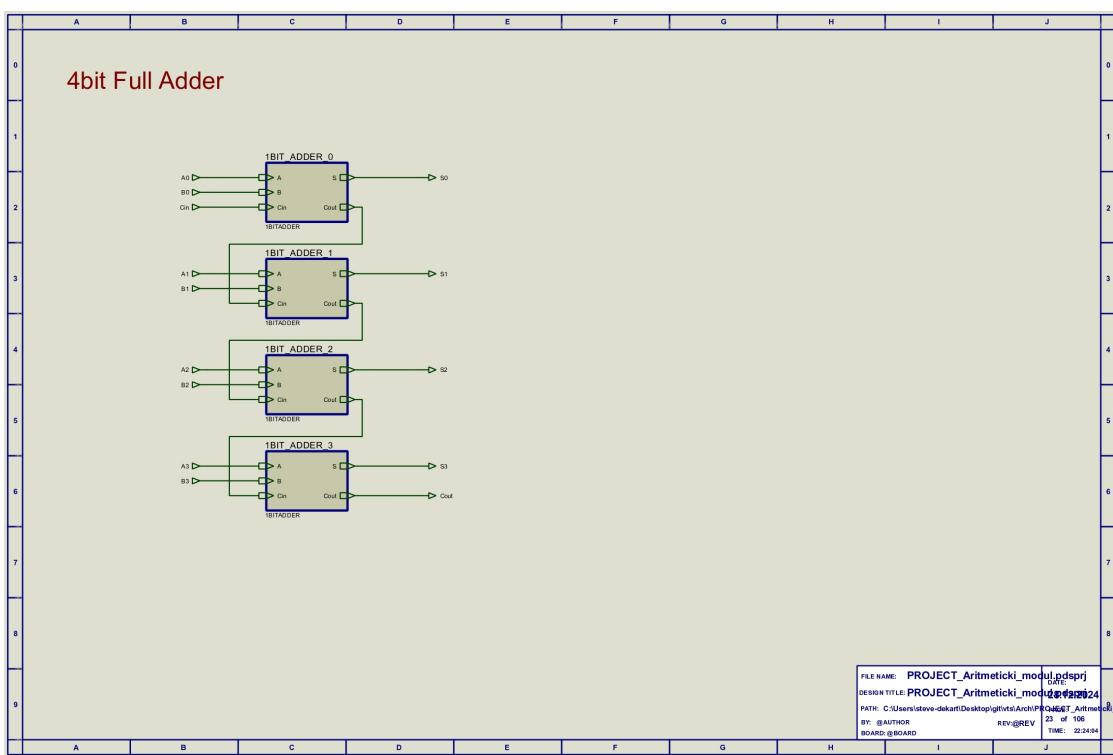
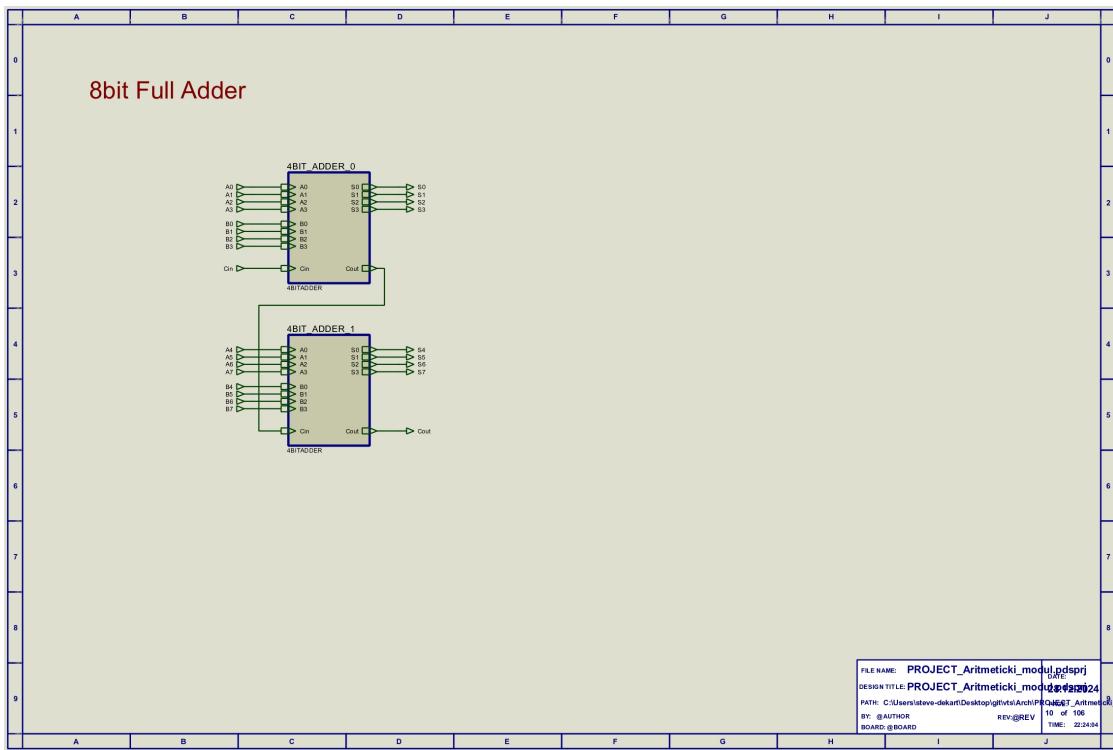
Slika 26. – Operacija  $A--$

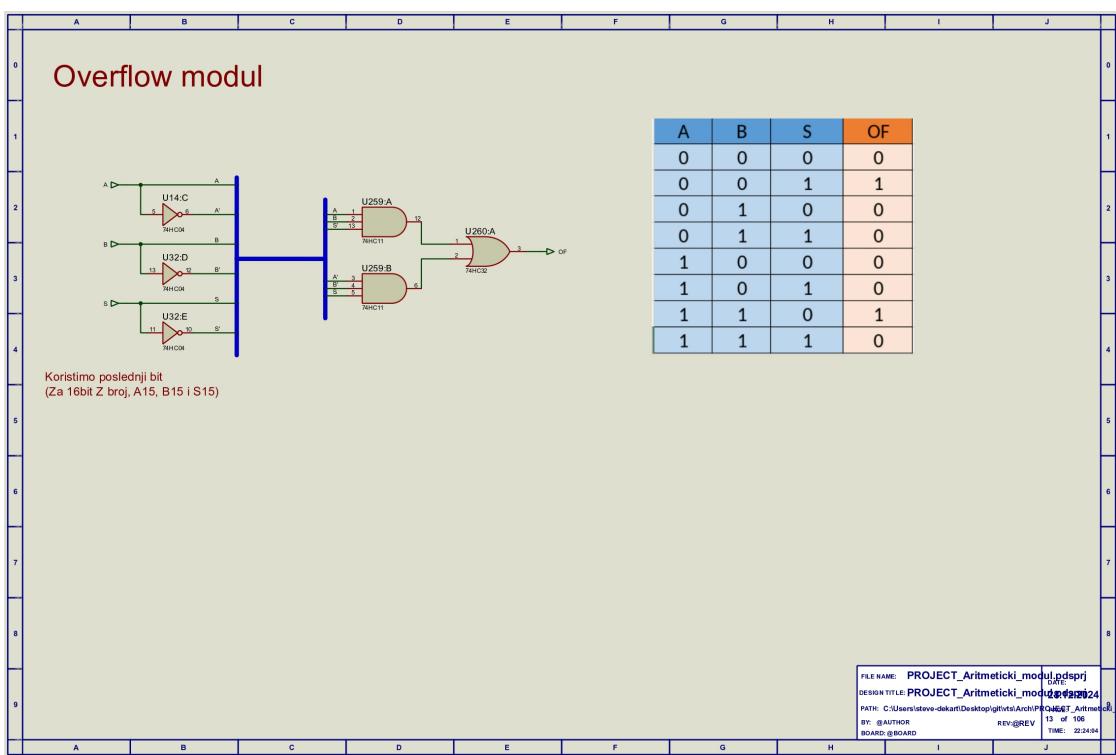
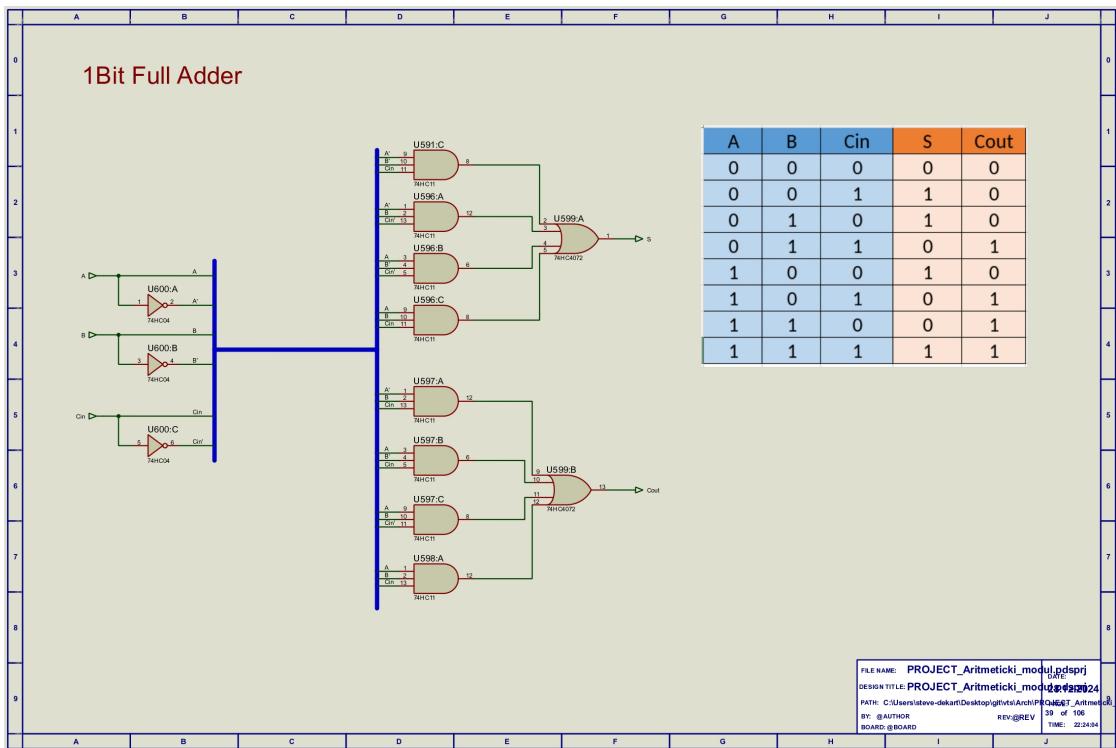
FILE NAME: P

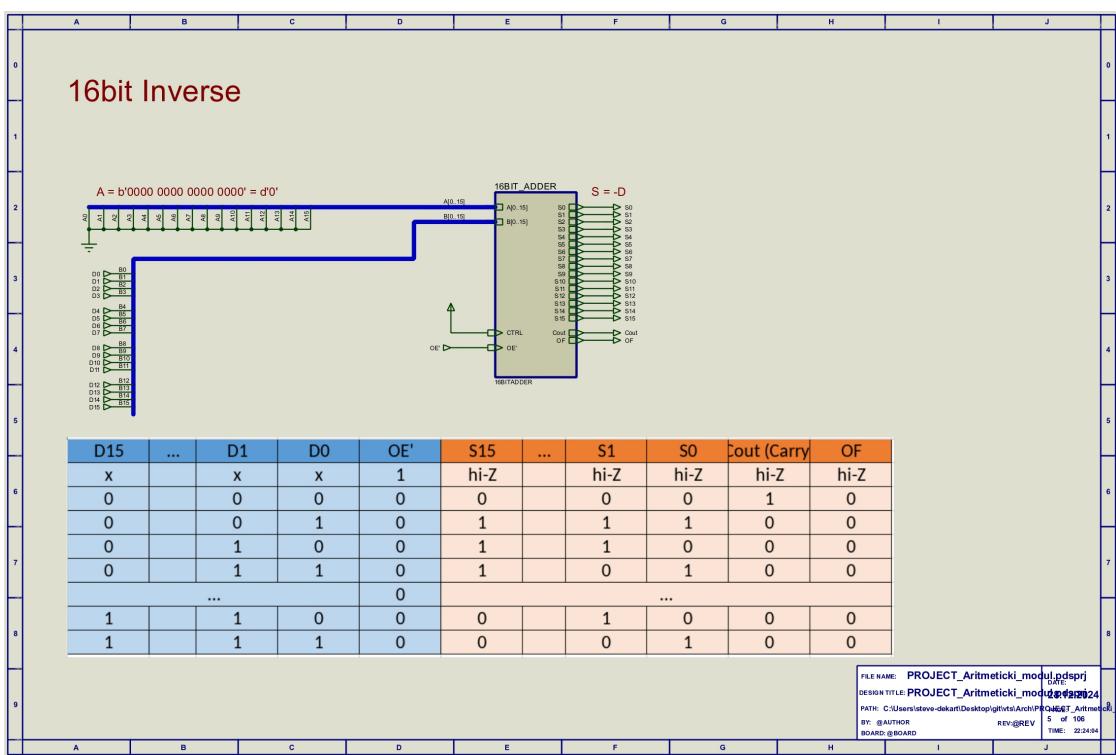
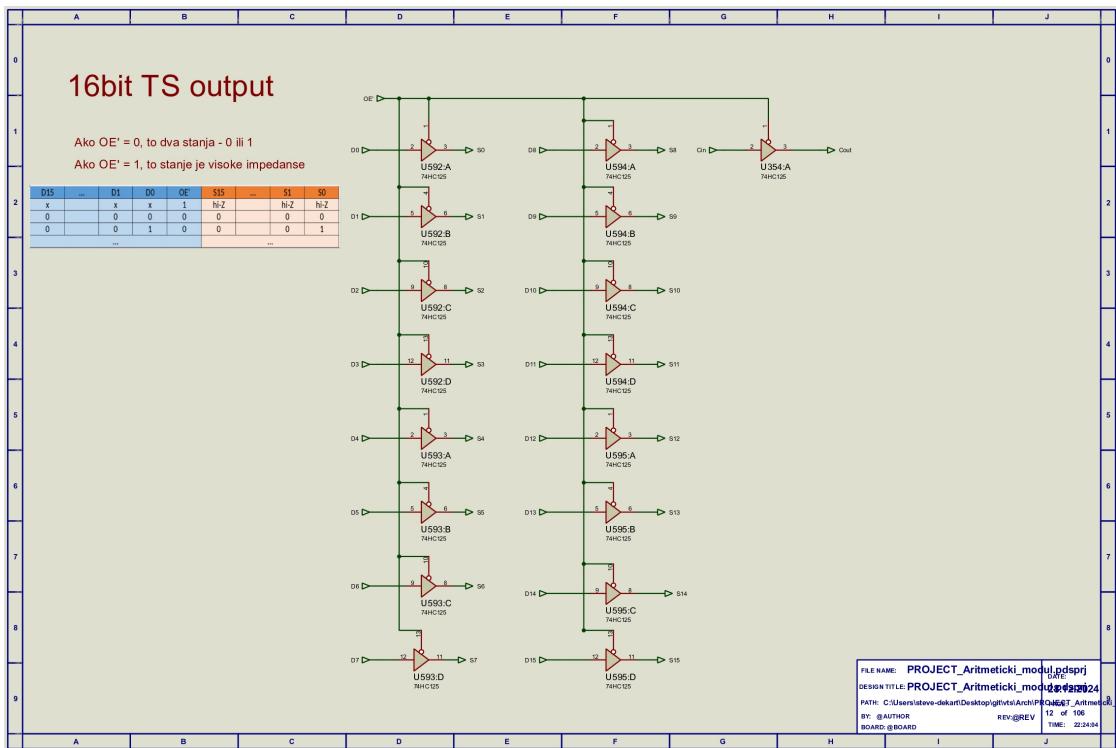
## 4. Prilog

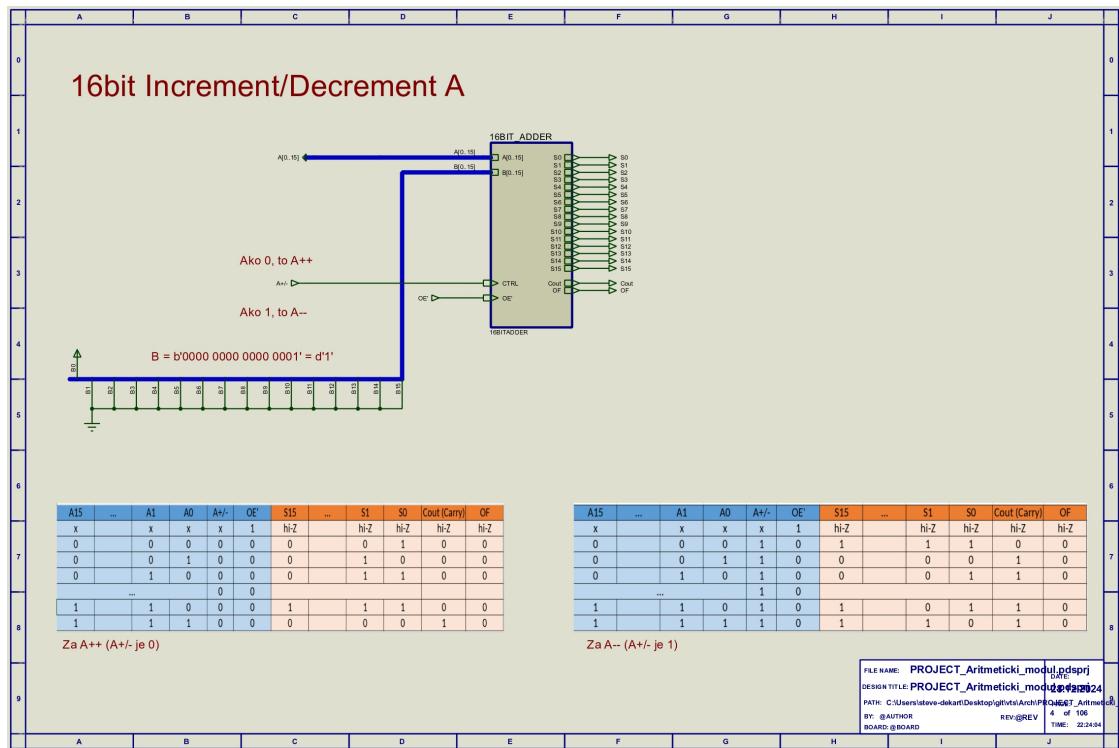












## **5. Literatura**

[W1] <http://www.csc.villanova.edu/~mdamian/Past/csc2400fa13/assign/ALU.html>

[W2] <https://www.geeksforgeeks.org/full-adder-in-digital-logic/>