

**»Mladi za napredek Maribora 2015«  
32. srečanje**

**MINECRAFT**

Raziskovalno področje Računalništvo

Raziskovalna naloga

PROSTOR ZA NALEPKO

Avtor: KEVIN KOS, DAMIJAN ROBNIK, JAN KRALJ

Mentor: IDA LOTRIČ

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA

**2015, Maribor**

## **Kazalo:**

1. Uvod .....	4
2. Osnovna logična vezja: .....	5
2.1. AND-IN logično vezje: .....	5
2.2. OR-ALI logično vezje: .....	6
2.3. NOT-NE logično vezje: .....	8
3. Sestavljena logična vezja: .....	9
3.1. NAND-NE IN logično vezje: .....	10
3.2. NOR-NE ALI logično vezje: .....	11
3.3. XOR logično vezje: .....	13
3.4. XNOR logično vezje:.....	15
4. Polovični seštevalnik (half adder):.....	17
5. Zaključek:.....	19
6. Viri: .....	20

## Kazalo slik:

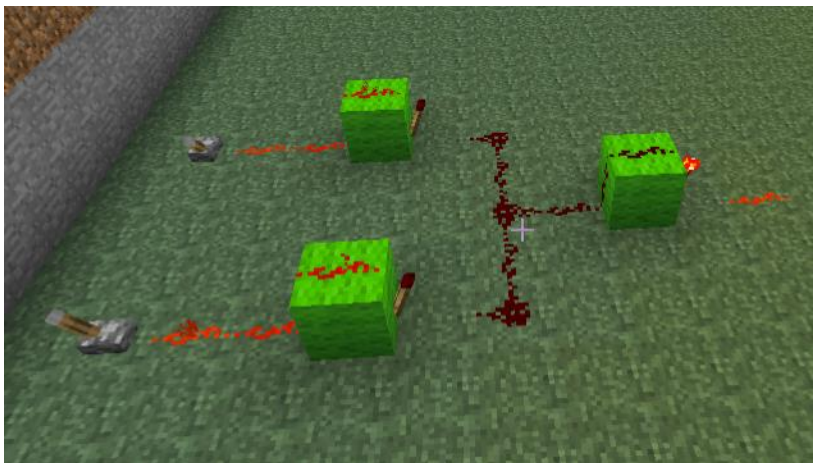
Slika 1: Primer NOR vezja v Minecraftu.....	4
Slika 2: Simbol za AND (IN) .....	5
Slika 3: AND (IN) vezje .....	5
Slika 4: AND (IN) vezje .....	6
Slika 5: AND (IN) vezje .....	6
Slika 6: Simbol za OR (ALI) .....	7
Slika 7: OR (ALI) vezje .....	7
Slika 8: OR (ALI) vezje .....	7
Slika 9: Simbol za NOT (NE) .....	8
Slika 10: NOT (NE) vezje .....	8
Slika 11: NOT (NE) vezje .....	9
Slika 12: NOT (NE) vezje .....	9
Slika 13: Simbol za NAND (NE-IN).....	10
Slika 14: NAND (NE IN) vezje .....	10
Slika 15: NAND (NE ALI) vezje.....	11
Slika 16: Simbol za NOR (NE ALI) .....	11
Slika 17: NOR (NE ALI) vezje .....	12
Slika 18: NOR (NE ALI) vezje .....	12
Slika 19: NOR (NE ALI) vezje .....	12
Slika 20: Simbol za XOR .....	13
Slika 21: XOR vezje .....	13
Slika 22: XOR vezje .....	14
Slika 23: XOR vezje .....	14
Slika 24: Simbol za XNOR.....	15
Slika 25: XNOR vezje .....	15
Slika 26: XNOR vezje .....	16
Slika 27: XNOR vezje .....	16
Slika 28: Logični diagram half adder-ja.....	17
Slika 29: Vezje half adder-ja .....	17
Slika 30: Vezje half adder-ja .....	18

# 1. UVOD

Računalništvo ima zelo velik pomen v današnjem času, vendar se večina ljudi tega sploh ne zaveda. Dan danes brez računalništva večina sistemov, npr. bančni, zdravstveni, šolski sploh ne bi delovali, saj so le-ti postali preveč odvisni od sodobnega računalništva. Zato smo hoteli z našo raziskovalno nalogo predstaviti računalništvo na preprost način, in sicer smo uporabili igro Minecraft, ki je zaslovela z njeno preprostostjo uporabe.

Minecraft je igra, v kateri lahko predstavimo vse naše ideje, ki so bile praktično neizvedljive ali bi za to porabili preveč časa (tudi pomemben faktor je, da je potrebno našo idejo prikazati čim bolj preprosto brez da bi izgubili tisto ciljno zasnov). Vsi objekti so izdelani iz kock, kar olajša kompleksnost določenih stvari.

Za predstavitev osnov računalništva smo si zamislili, da ustvarimo in prikažemo delovanje osnovnih logičnih vezij (AND-IN, OR-ALI in NOT-NE), sestavljenih logičnih vezij (NAND, NOR, XOR in XNOR) in polovični seštevalnik (Half adder), saj osnovne aritmetične operacije delujejo na principu teh vezij.



*Slika 1: Primer NOR vezja v Minecraftu*

## 2. OSNOVNA LOGIČNA VEZJA:

Osnovna logična vezja so AND (IN), OR (ALI) in NOT (NE), s katerimi lahko izvajamo osnovne logične operacije. Te osnovna vezja lahko združimo in dobimo sestavljena vezja, o katerih bomo govorili kasneje.

### 2.1. AND-IN logično vezje:

AND-IN je osnovna logična funkcija, ki implementira logično konjunkcijo. Na izhodu dobimo 1 samo takrat, če sta oba dva vhoda 1. Kadar pa je eden ali noben izmed vhodov 1, pa je izhod 0. To si lahko predstavljamo tudi tako, da AND funkcija išče minimum med dvema binarnima številoma, zaradi česar je izhod vedno 0 razen če so vsi vhodi 1. Primer pravilnostne tabele (truth table) za 2 vhoda in 1 izhod:

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



Slika 2: Simbol za AND (IN)



Slika 3: AND (IN) vezje



Slika 4: AND (IN) vezje

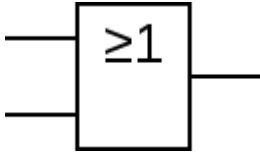


Slika 5: AND (IN) vezje

## 2.2. OR-ALI logično vezje:

OR-ALI je logično vezje, katere osnovna funkcija je logična disjunkcija. Izhod je true (1) takrat, kadar je eden ali obadva vhoda true (1). Če pa nobeden izmed vhodov ni enak 1, je izhod false (0). V drugačnem smislu ta funkcija išče maksimum med dvema binarnima številoma. Primer pravilnostne tabele (truth table) za 2 vhoda in 1 izhod:

INPUT		OUTPUT
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1



Slika 6: Simbol za OR (ALI)



Slika 7: OR (ALI) vezje

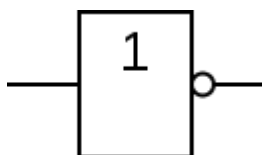


Slika 8: OR (ALI) vezje

## 2.3. NOT-NE logično vezje:

NOT-NE je logično vezje, ki implementira negacijo, kar pomeni, da dobimo na izhodu 1 le takrat, če je na vhodu 0, kadar pa je vhod 1, je izhod 0. Primer pravilnostne tabele (truth table) za 1 vhod in 1 izhod:

INPUT	OUTPUT
A	NOT A
0	1
1	0



Slika 9: Simbol za NOT (NE)



Slika 10: NOT (NE) vezje





*Slika 11: NOT (NE) vezje*



*Slika 12: NOT (NE) vezje*

### 3. SESTAVLJENA LOGIČNA VEZJA:

Sestavljena logična vezja so NAND (NE IN), NOR (NE ALI), XOR (antivalenca) in XNOR (ekvivalenca). Ta vezja so bolj kompleksnejša in se uporabljajo za bolj zahtevne logične operacije.

#### 3.1. NAND-NE IN logično vezje:

NAND-NE IN je logična funkcija, ki proizvede izhod, kateri je lahko false (0) samo takrat, če so vsi vhodi true (1); torej je izhod komplementaren AND funkciji. Če pa je eden ali obadva vhoda 0, pa je izhod 1.

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



Slika 13: Simbol za NAND (NE-IN)



Slika 14: NAND (NE IN) vezje

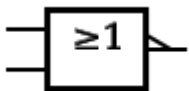


Slika 15: NAND (NE ALI) vezje

### 3.2. NOR-NE ALI logično vezje:

NOR-NE ALI je logično vezje, ki implementira logični NOR. Logično vezje deluje tako, da dobimo na izhodu true (1) samo takrat, če sta oba vhoda false (0), v nasprotnem primeru pa je na izhodu false (0). NOR je rezultat negacija OR operatorja. Lahko si ga tudi predstavljamo kot AND vrata z vsemi vhodi invertiranimi. Primer pravilnostne tabele (truth table) za 2 vhoda in 1 izhod:

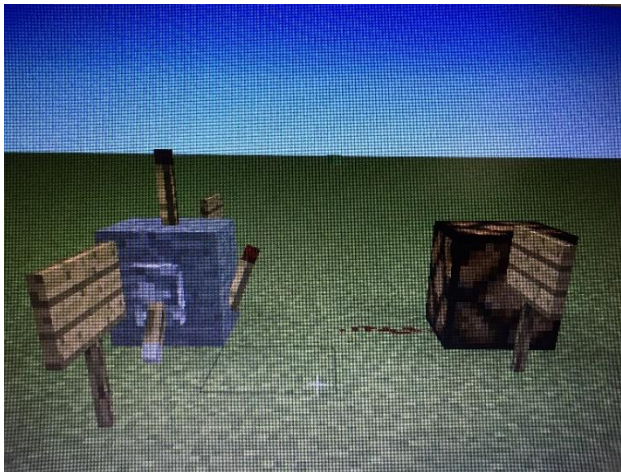
INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0



Slika 16: Simbol za NOR (NE ALI)



Slika 17: NOR (NE ALI) vezje



Slika 18: NOR (NE ALI) vezje

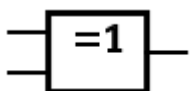


Slika 19: NOR (NE ALI) vezje

### 3.3. XOR logično vezje:

XOR (ekvivalenca) je logično vezje, ki implementira ekskluzivni NE, kar pomeni, da je na izhodu 1, če je samo eden izmed vhodov true (1). Če pa sta oba vhoda false (0) ali true (1), v tem primeru je izhod false (0). XOR predstavlja funkcijo neenakosti. Dober primer za zapomnjenje te funkcije je »eden ali drugi vendar ne obadva«. Primer pravilnostne tabele (truth table) za 2 vhoda in 1 izhod:

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Slika 20: Simbol za XOR



Slika 21: XOR vezje





*Slika 22: XOR vezje*

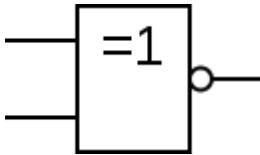


*Slika 23: XOR vezje*

### 3.4. XNOR logično vezje:

XNOR (antivalenca) je logično vezje, katere funkcija je logični komplement XOR vrat. Dva vhodna vezja implementira logično enakost, kar pomeni, da kadar sta oba vhoda enaka, dobimo na izhodu 1, v nasprotnem primeru če je eden vendar ne oba vhoda 1, dobimo na izhodu 0. Primer pravilnostne tabele (truth table) za 2 vhoda in 1 izhod:

Input		Output
A	B	
0	0	1
0	1	0
1	0	0
1	1	1



Slika 24: Simbol za XNOR



Slika 25: XNOR vezje



Slika 26: XNOR vezje



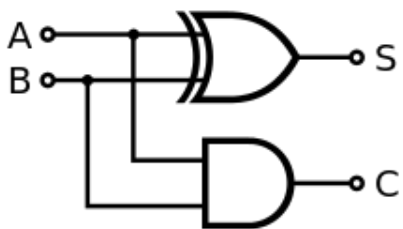
Slika 27: XNOR vezje



## 4. POLOVIČNI SEŠTEVALNIK (half adder):

Polovični seštevalnik (half adde) sešteva dve enomestni binarni števili A in B. Ima dva izhoda, suma (S) in carry-prenos (C). Carry signal predstavlja prenos v naslednjo številko več številčnega seštevanja. Najpreprostejši half-adder design, pod besedilom, vsebuje XOR vrata za S in AND vrata za C. Half adder sešteva dva vhodna bita in generira carry in sumo, ki sta dva izhoda pri half adder-ju. Pravilnostna tabela (truth table) za half adder je:

Inputs		Outputs	
A	B	C	S
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0



Slika 28: Logični diagram half adder-ja



Slika 29: Vezje half adder-ja



*Slika 30: Vezje half adder-ja*

## **5. ZAKLJUČEK:**

Vsa osnovna logična in sestavljena vezja ter half-adder (polovični seštevalnik) smo uspešno prikazali, in sicer njihovo delovanje, kakršno je v resničnosti, tudi v igri Minecraft. Z našo raziskovalno nalogo smo marsikateremu približali osnovne pojme računalništva, ki so mu bile pred tem neznane in celo pretežke za razumevanje. Minecraft se je izkazal kot odlična platforma za prikaz vseh raznih vezij, kar si jih lahko zamislimo, in pri tem ne potrebujemo sestavljati pravih vezij v resničnosti, vendar potrebujemo le en načrt za tisto določeno vezje, in smo tudi ugotovili, da smo prihranili veliko časa z izdelavo vezij v samem Minecraftu.

Moram pa izpostaviti našega mentorja, ki nam je zelo pomagal pri izdelavi te raziskovalne naloge, ki je bila kar en zalogaj za uspešno dokončanje .

## 6. VIRI:

[http://en.wikipedia.org/wiki/AND\\_gate](http://en.wikipedia.org/wiki/AND_gate) (9.2.2014)

[http://en.wikipedia.org/wiki/OR\\_gate](http://en.wikipedia.org/wiki/OR_gate) (9.2.2014)

[http://en.wikipedia.org/wiki/Inverter\\_%28logic\\_gate%29](http://en.wikipedia.org/wiki/Inverter_%28logic_gate%29) (9.2.2014)

[http://en.wikipedia.org/wiki/NAND\\_gate](http://en.wikipedia.org/wiki/NAND_gate) (9.2.2014)

[http://en.wikipedia.org/wiki/NOR\\_gate](http://en.wikipedia.org/wiki/NOR_gate) (9.2.2014)

[http://en.wikipedia.org/wiki/XOR\\_gate](http://en.wikipedia.org/wiki/XOR_gate) (9.2.2014)

[http://en.wikipedia.org/wiki/XNOR\\_gate](http://en.wikipedia.org/wiki/XNOR_gate) (9.2.2014)

[http://en.wikipedia.org/wiki/Adder\\_%28electronics%29](http://en.wikipedia.org/wiki/Adder_%28electronics%29) (9.2.2014)