**4-bit Fisherman**

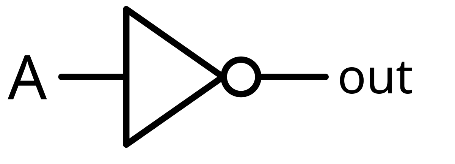
Nama: Aryo Bama Wiratama

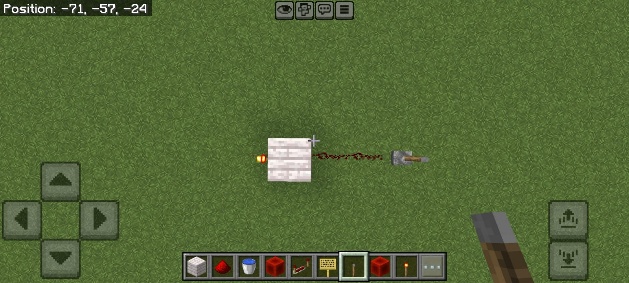
NIM: 13523088

1. Gate

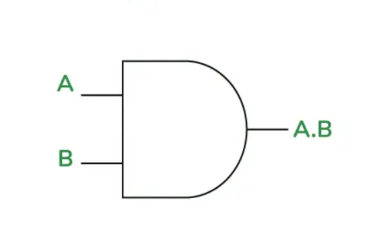
Program yang digunakan untuk menyimulasikan adalah game minecraft dengan memanfaatkan *redstone system* di dalam game. Sebelum semuanya dimulai, terdapat gate dasar yang akan digunakan untuk menyimulasikan CPU. Berikut beberapa gate yang akan dipakai serta implementasinya dalam game Minecraft:

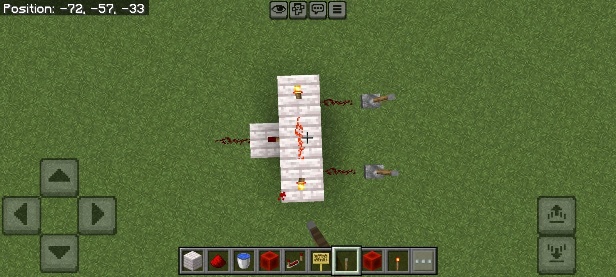
1. NOT Gate



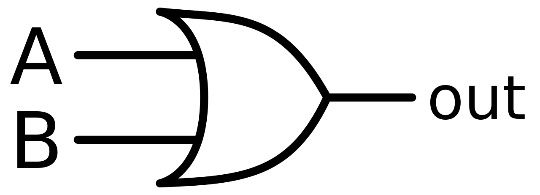


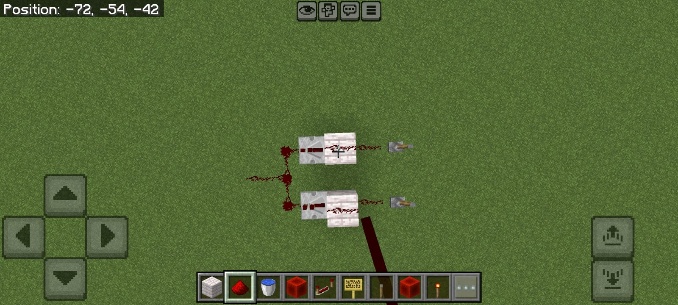
1. AND Gate



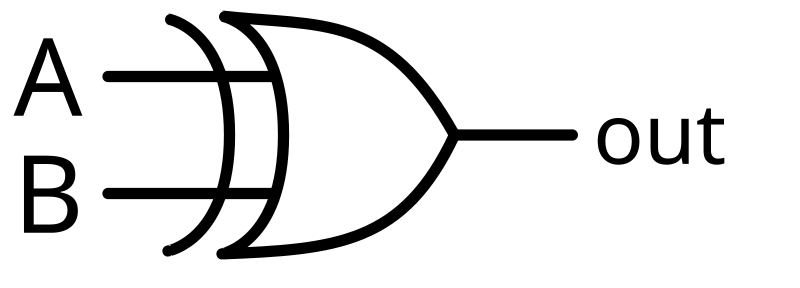


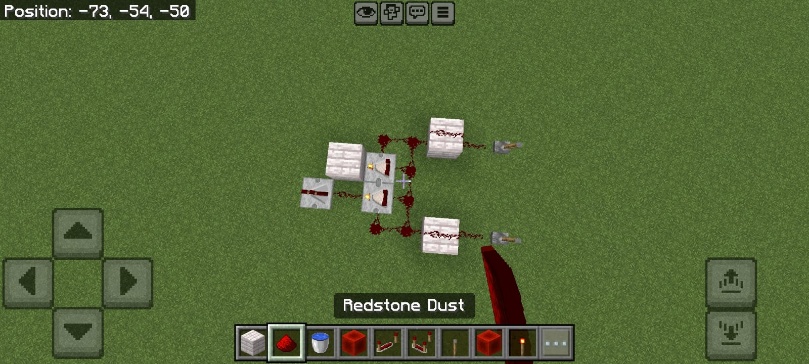
1. OR Gate





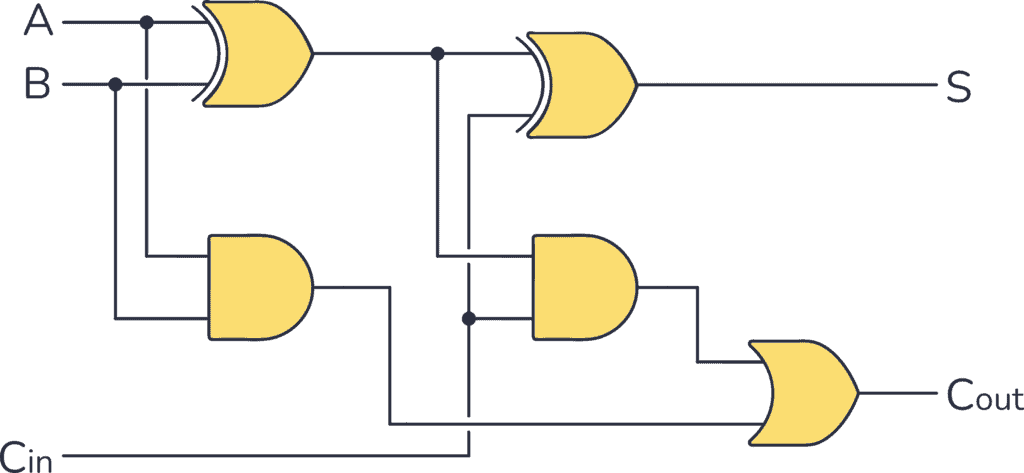
1. XOR Gate



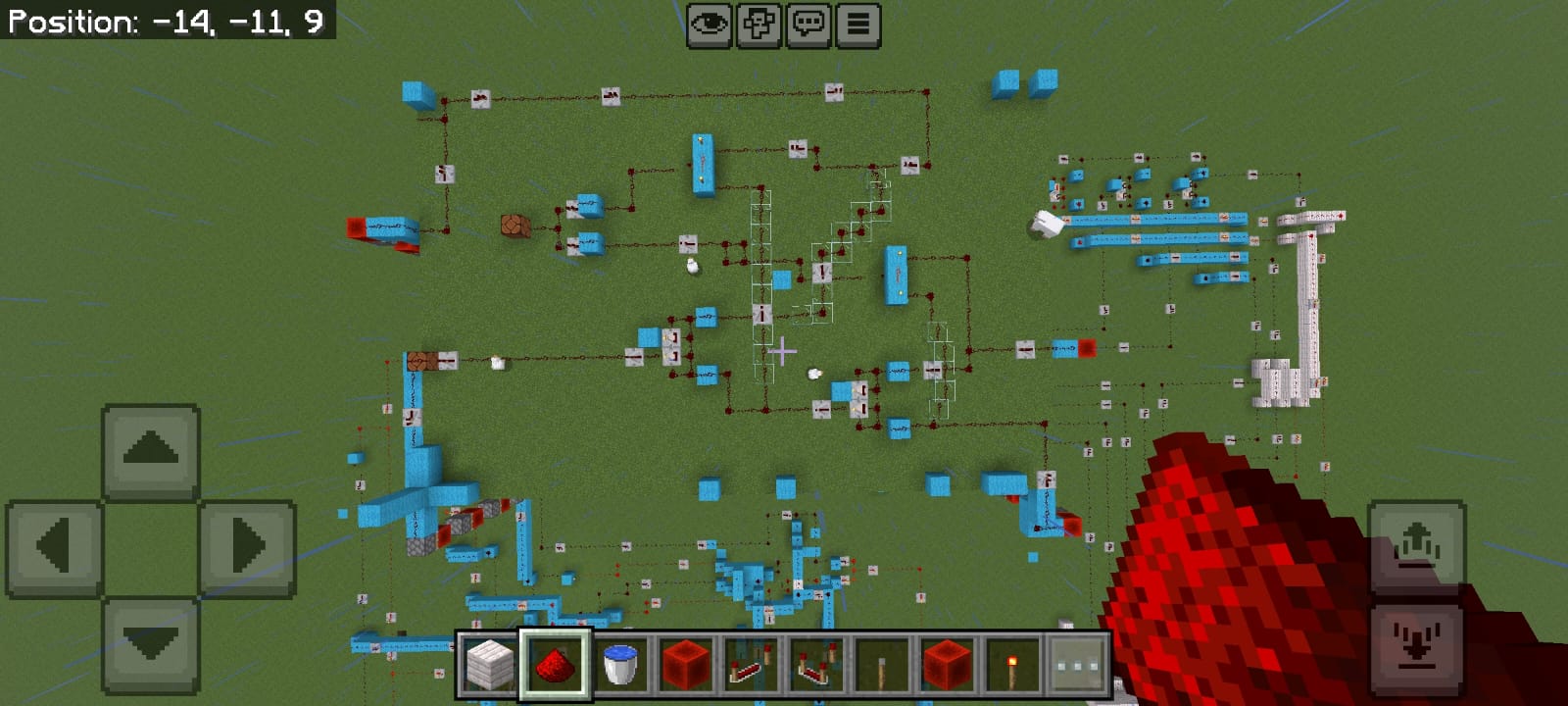


1. Full Adder

**Full adder** adalah rangkaian logika digital yang digunakan untuk menjumlahkan tiga bit input, yaitu dua bit data (A dan B) serta satu bit *carry-in* dari penjumlahan sebelumnya. Full adder menghasilkan dua output, yakni *sum* (hasil penjumlahan bit) dan *carry-out* (hasil bawaan yang dikirim ke penjumlahan bit berikutnya). Untuk melakukan penjumlahan bilangan biner lebih dari 1 bit, beberapa full adder dapat disusun secara berantai, membentuk **ripple-carry adder**. Misalnya, dalam penjumlahan 4 bit, digunakan empat full adder yang masing-masing menangani satu posisi bit dari LSB (Least Significant Bit) hingga MSB (Most Significant Bit). *Carry-out* dari satu full adder akan menjadi *carry-in* untuk full adder berikutnya. Dengan cara ini, dua bilangan biner 4 bit, misalnya 1011 (11 desimal) dan 0110 (6 desimal), dapat dijumlahkan bit per bit dari kanan ke kiri: mulai dari LSB dengan memperhitungkan carry, hingga akhirnya diperoleh hasil 10001 (17 desimal). Full adder dapat disusun dari logic gate dasar. Berikut rangkaian full Adder.



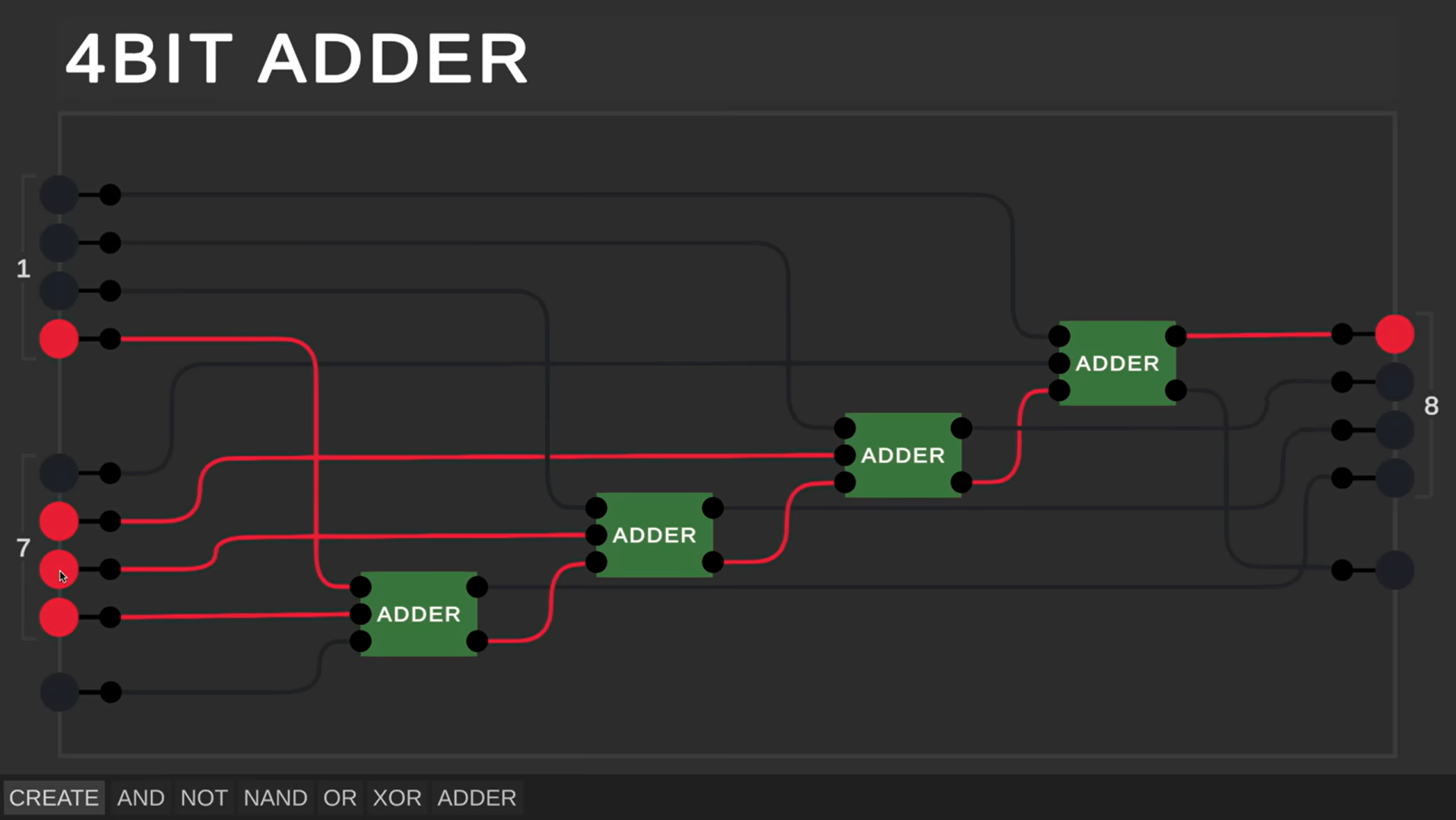
Berikut implementasinya dalam minecraft



Untuk menyimulasikan 4 bit dibutuhkan 4 full adder. Oleh karena itu, penulis membuat 4 full adder yang disusun



Berikut ilustrasi cara kerja 4 full adder dalam menjumlahkan 2 input 4 bit:

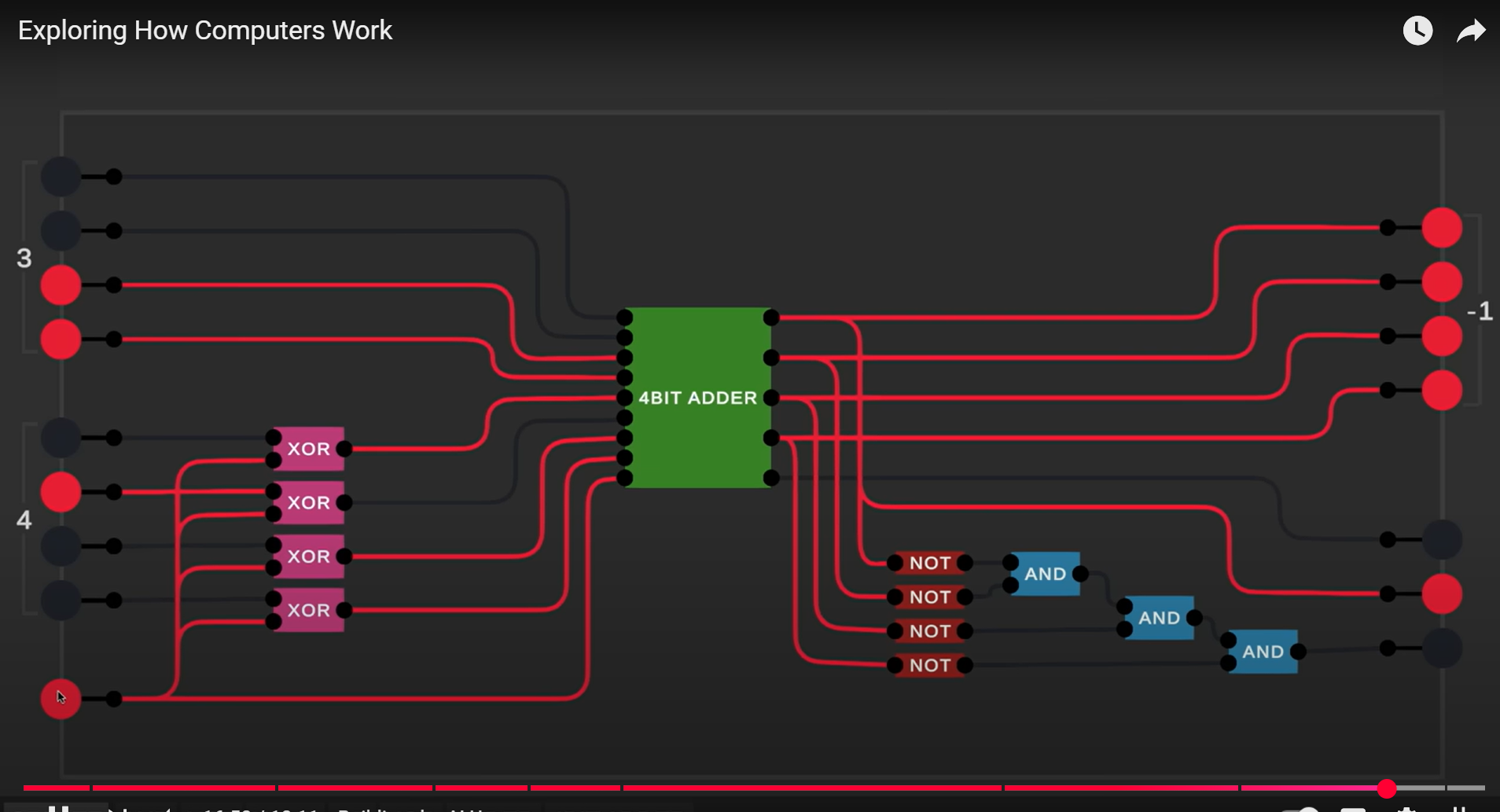


(Sumber: <https://www.youtube.com/watch?v=QZwneRb-zqA>)

1. Operasi Pengurangan

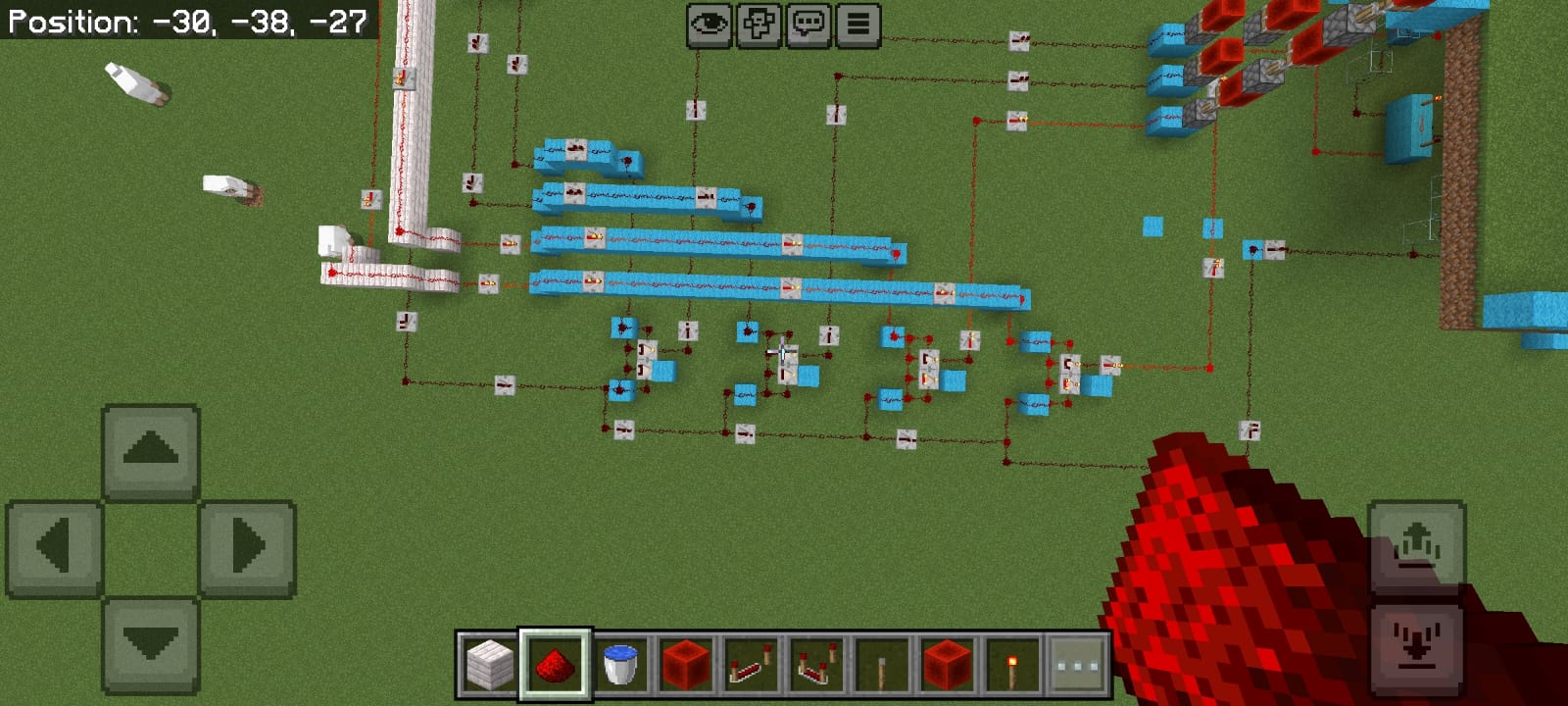
Untuk membaca bilangan negatif, komputer biasanya menggunakan reperesentasi two complements. Komputer memanfaatkan MSB (Most significant bit) atau bit yang berada di paling depan. Bit tersebut disebut sebagai sign. Sign yang aktif akan dihitung sebagai angka negatif. Terdapat langkah yang dibutuhkan dalam membaca bit negatif dalam representasi two complements. Misal diberi angka 1100, bit tersebut memiliki sign yang aktif sehingga akan dibaca sebagai bit negatif. Pertama balik bit tersebut sehinnga menjadi 0011, kemudian tambah 1 bit tersebut sehingga menjadi 0100. Apabila dikonvesi menjadi desimal bit tersebut adalah 4 sehingga 1100 adalah representasi dari -4

Kita dapat memanfaatkan hal tersebut untuk melakukan pengurangan. Misal terdapat operasi A – B. Untuk melakukan hal tersebut kita lakukan aturan two complements pada B. Berikut ilustrasinya



(Sumber: <https://www.youtube.com/watch?v=QZwneRb-zqA>)

Berikut implementasi pada minecraft

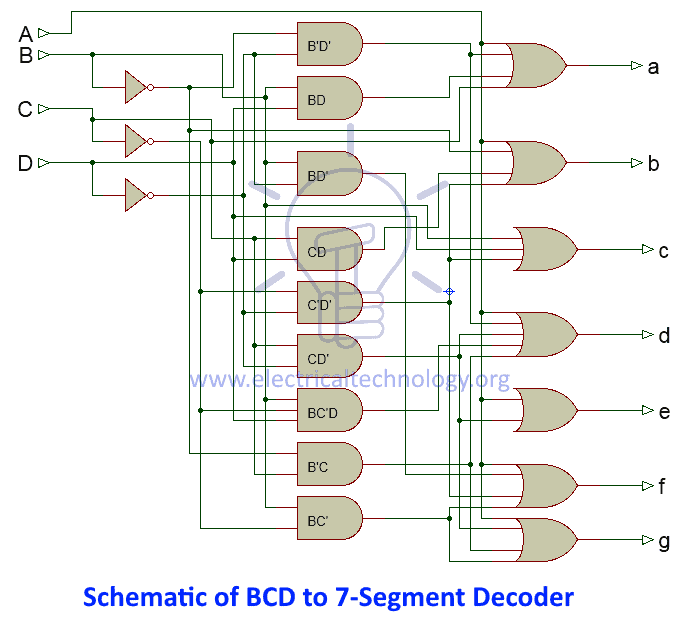


Gambar di atas mengimplementasikan XOR gate pada input B sebelum masuk ke Full Adder

1. Seven segment display

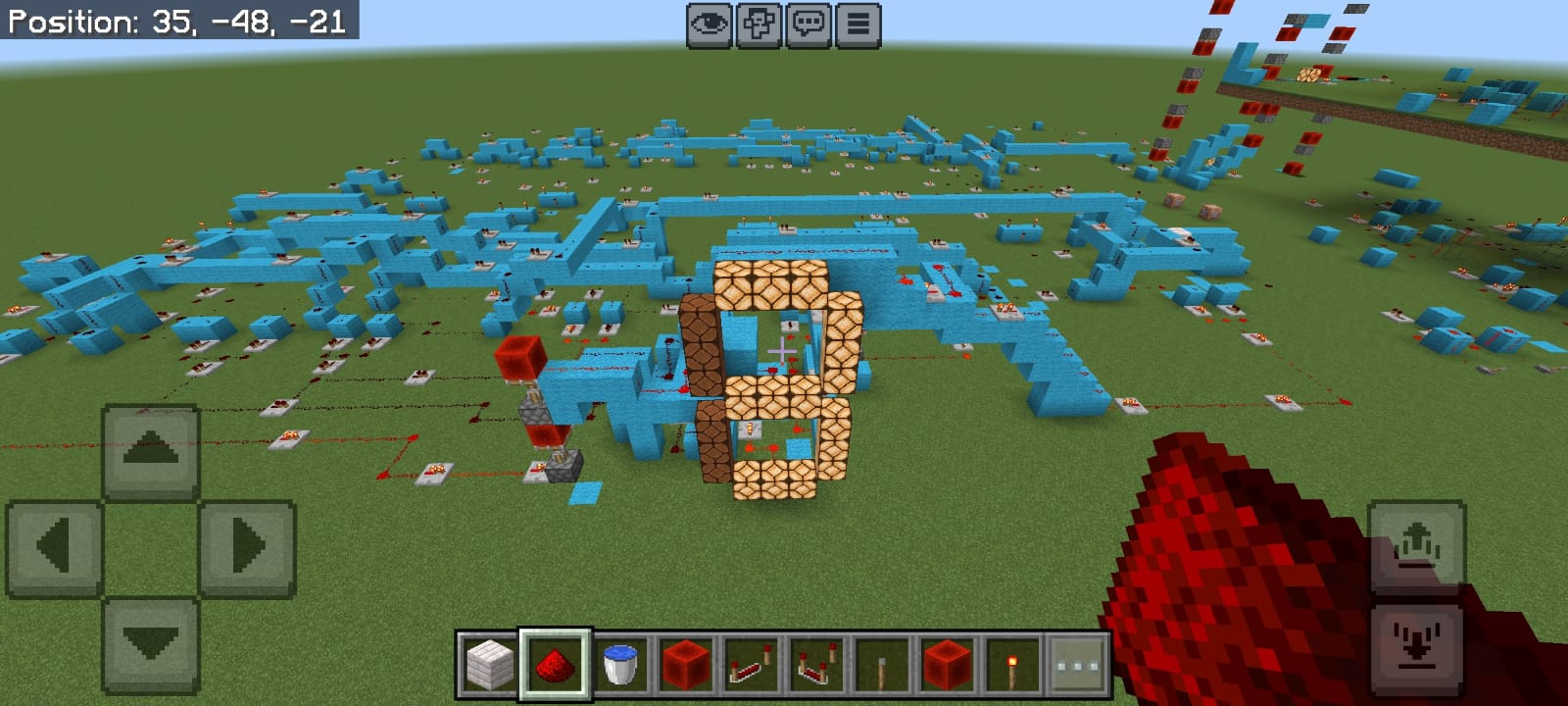
Seven-segment display adalah perangkat elektronik yang digunakan untuk menampilkan angka desimal dari 0 hingga 9. Cara kerjanya adalah dengan mengonversi input biner (biasanya dari 4-bit, yang dikenal sebagai Binary-Coded Decimal atau BCD) menjadi sinyal listrik yang mengaktifkan segmen-segmen tertentu. Setiap segmen diberi label dari A sampai G, ditambah titik desimal (DP). Untuk menampilkan angka, setiap segmen harus dihidupkan (menyala) atau dimatikan (tidak menyala) sesuai dengan pola yang diinginkan. Proses konversi ini biasanya dilakukan oleh sebuah dekoder, seperti IC 7447 (untuk *common-anode*) atau IC 7448 (untuk *common-cathode*), yang berfungsi sebagai "penerjemah." Dekoder ini mengambil 4-bit BCD sebagai input, lalu secara otomatis menghasilkan output 7-bit yang sesuai untuk mengontrol segmen-segmen pada display, memastikan pola yang benar untuk setiap angka. Misalnya, untuk menampilkan angka "1", dekoder akan mengaktifkan segmen B dan C, sementara segmen lainnya tetap mati. Dengan cara ini, display dapat menampilkan angka-angka yang dapat dibaca oleh manusia dari data biner.

Dekoder dalam seven segment display juga terdiri dari beberapa logic gate. Logic gate tersebut tergantung pada kombinasi segment yang diperlukan untuk menyalakan lampu pada nilai tertentu. Berikut dekoder yang saya gunakan



Logic gate di atas didapat dari penyederhanaan aljabar boolean yang menggunakan beberapa metode peta karnaugh. Penjelasan terkait itu, dapat dilihat pada link berikut <https://www.electricaltechnology.org/2018/05/bcd-to-7-segment-display-decoder.html>

Berikut implementasinya



Rangkaian yang berantakan di belakang segment tersebut merupaka implementasi dari logic gate yang sudah ditentukan di atas. Pada seven segment display yang saya buat terdapat kekurangan, yaitu seven segment tidak menghandle angka negatif sehingga output yang menghasilkan angka negatif dapat salah. Walaupun begitu, apabila dilihat dari output bit dari 4 full adde,r operasi yang menghasilkan angka negatif akan sesuai dengan representasi two complement, misal operasi (-4)- 4 akan menghasilkan -8 di mana dalam two complement hasilnya akan 1 0 0 0. Namun, karena seven segment tidak menghandle negatif maka pada output layar akan menampilkan 8

1. Input

Interface untuk input yang dibuat sangatlah sederhana. Pengguna dapat memasukkan angka dalam representasi bit. Berikut gambar input

