

Referensi simple : <https://youtu.be/Z5JC9Ve1sfI?si=CYrbduGBeHIWwC7Z>

CPU Merupakan komponen vital dalam suatu komputer, karena memungkinkan komputer untuk “Berpikir” yakni melakukan komputasi/perhitungan sesuai perintah yang diberikan dengan kecepatan yang sangat-sangat cepat,

CPU Modern menggunakan siklus “Fetch Decode Execute” dimana terdapat kurang lebih 3 step yang diulang-ulang oleh CPU.

- Fetch
 - CPU melakukan read ke memory pada posisi sesuai program counter dan mengingat nya
- Decode
 - Informasi yang didapatkan pada fetch diubah jadi suatu perintah untuk dilakukan mesin (misalnya ADD)
- Execute
 - Perintah dilakukan oleh CPU dan hasilnya disimpan.

Dengan definisi ini, untuk membuat simulasi CPU 4 bit simpel, akan diperlukan komponen-komponen berikut

- Memory
- Program Counter
- Control Unit
- ALU (Dalam kasus ini cukup pertambahan dan pengurangan)
- Register

Perintah yang akan dibuat adalah : ~~Read from address, Write to address~~, Addition, dan Subtraction. Namun, karena batasan 4 bit, terdapat adaptasi yang perlu dilakukan. Untuk memungkinkan penggunaan penuh seluruh 4 bit untuk kalkulasi :

- Untuk semua perintah, akan dibagi menjadi dua yaitu Perintahnya, dan Angkanya. Sehingga jika address X misalnya perintah untuk ADD, maka X+1 adalah angka yang ingin ditambahkan

Notasi XXXX | YYYY berarti ada “value” di address setelah perintah yang akan digunakan
Notasi XXXX berarti hanya perlu itu saja

~~00XX | YYYY : READ FROM (YYYY) TO REGISTER XX~~

~~01XX | YYYY : WRITE AT (YYYY) FROM REGISTER XX~~

~~1000 | YYYY : JUMP TO YYYY~~

1100 | YYYY : ADDITION (REGISTER 0 + YYYY)

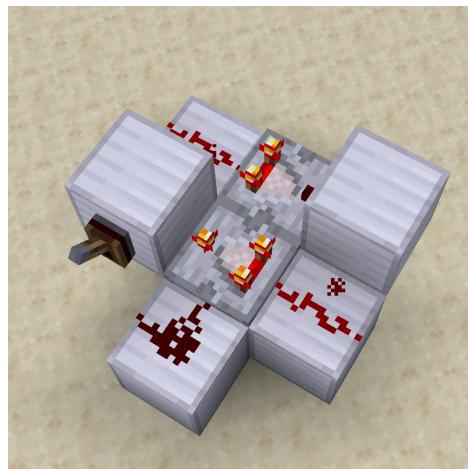
1101 | YYYY : SUBTRACTION (REGISTER 0 - YYYY)

Memory :

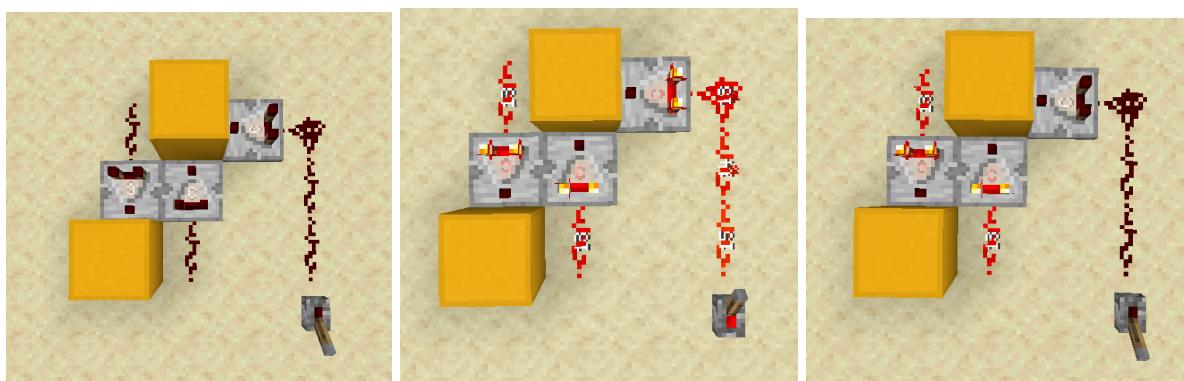
Memory digunakan untuk menaruh/menuliskan perintah-perintah yang akan dijalankan oleh CPU, serta menuliskan hasil atau value pada suatu register untuk memungkinkan berbagai macam perhitungan kompleks

Redstone di Minecraft bisa memiliki value dari 0 sampai 15, angka ini kebetulan sama dengan range dari hexadecimal 1 angka (0-F) atau setara dengan 4 bit. Penyimpanan data 4 bit bisa dilakukan dengan mekanisme yang cukup kecil. Mengambil referensi dari link berikut :

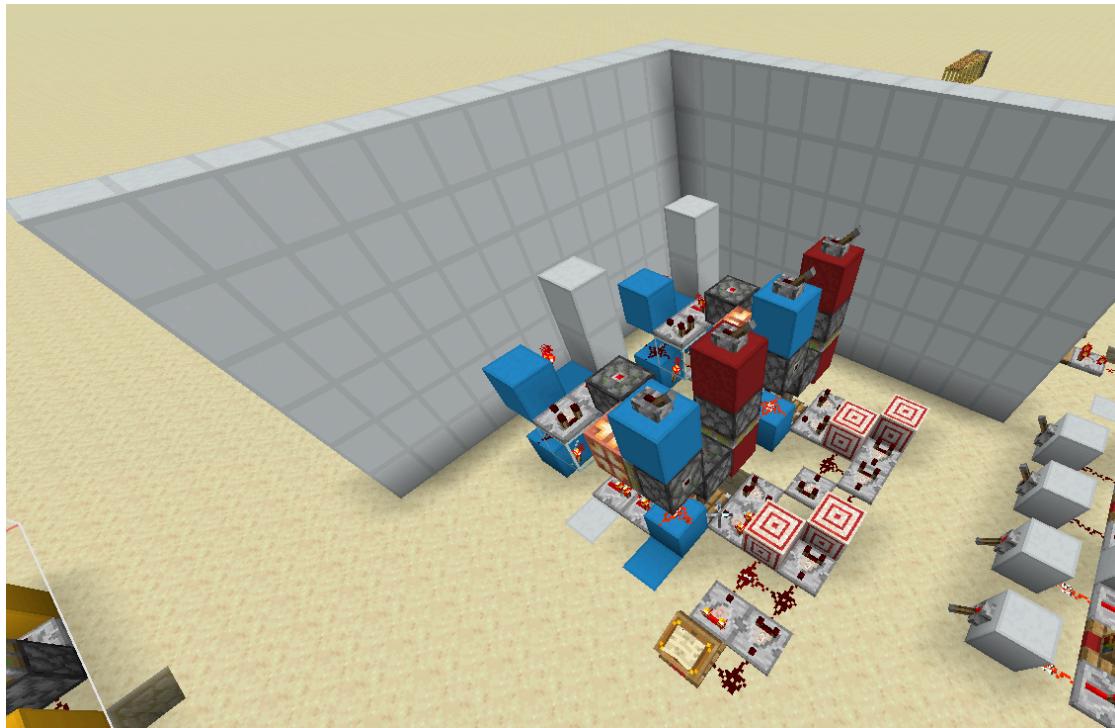
https://www.reddit.com/r/redstone/comments/ov3pe9/basiclly_its_a_memory_cell_that_locks_only_if/



Comparator memiliki sifat khusus yakni meneruskan sinyal redstone sesuai input nya, sehingga dapat digunakan untuk “menyimpan” informasi 4 bit



Dengan pengembangan lebih lanjut, dapat dibuat suatu sistem memory dimana suatu rangkaian memory dapat dihubungkan kepada satu kabel/jalur I/O dan memungkinkan untuk hanya mewrite ke suatu cell atau membaca dari suatu cell



Tuas pada jalur biru untuk write, tuas pada jalur merah untuk read

Design ini memungkinkan untuk satu cell memory berdempetan dengan cell lain nya. Namun memiliki kekurangan yaitu adanya waktu yang diperlukan untuk sinyal mencapai dari ujung ke ujung. Karena untuk menghubungkan satu cell ke cell sebelah memerlukan 1 comparator, untuk setiap jarak dari sumber/output akan memakan 1 tick (1 detik adalah 20 tick dalam minecraft pada kondisi normal/default). Sehingga jika misalnya terdapat 20 memory cell, maka cell di ujung dapat memerlukan waktu 1 detik sebelum informasi bisa sampai atau datang

PROGRAM COUNTER :

Program counter merupakan register yang digunakan untuk mengatur langkah/steps dalam CPU. Program Counter akan menunjuk kepada memory apa yang sedang dibaca, dan ketika cycle sudah selesai akan increment (+1) untuk melanjut ke bagian berikutnya.

Ketika ada perintah jump, maka destinasi dari jump akan menjadi alamat baru program counter, "mengubah" alur eksekusi program.

Setelah beberapa pencobaan saya berhasil membuat *contraption* redstone sebagai berikut :



Tombol di bagian bawah merupakan tombol untuk increment, sedangkan redstone diatas digunakan untuk mereset program counter



Reset (set semua ke 0)

Untuk merubah Program Counter, dapat dilakukan dengan pertama mereset PC, lalu memasukkan langsung value yang diinginkan.

Muneki

13522130



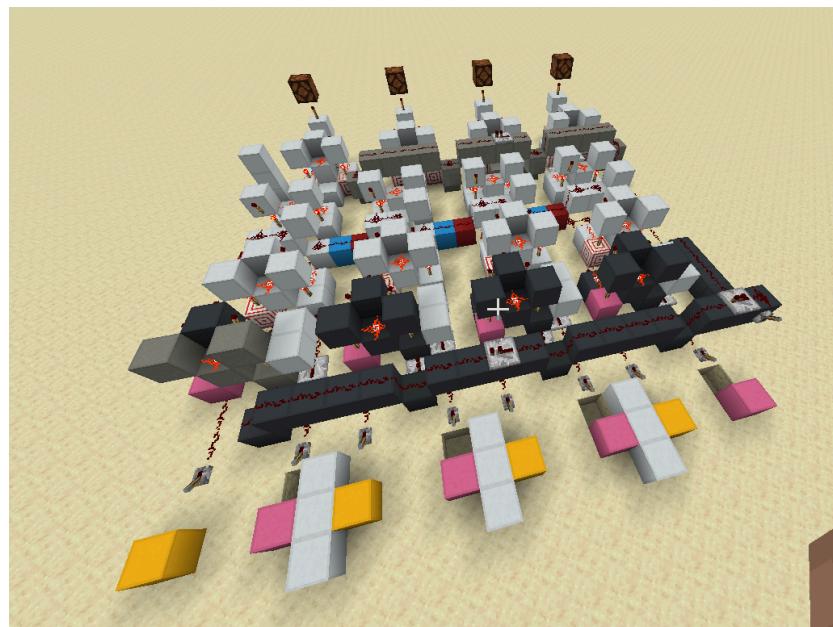
Set value tertentu

ALU :

Untuk rangkaian penambahan dan pengurangan, menggunakan referensi video berikut :

binary subtraction - Minecraft Java/Bedrock . Sebagai tambahan sistem ini merupakan sistem unsigned (bit paling kiri positif. Bukan negatif). Sistem ini juga belum memiliki handling untuk overflow.

Pada simulasi ini, ALU dihubungkan ke Register dan Input dari read memory. Namun penulisan ke Register dikendalikan oleh Control Unit.



Jalur kuning untuk A, jalur Pink untuk B, jalur hitam untuk mengubah mode menjadi pengurangan ($A + B \rightarrow A - B$).



Versi 1, Register yang langsung terhubung ke ALU

CONTROL UNIT

Komponen dalam simulasi ini yang paling kompleks, Control Unit bertugas untuk men Decode (pada simulasi ini) perintah yang di read, dan mengeksekusi nya.

Karena batasan 4 bit, setelah men-decode perintah, PC akan maju 1 untuk menerima input angka untuk operasi.



Tombol tombol yang digunakan untuk mengontrol CPU.

Komponen yang utama dalam Control Unit terdapat pada garis biru muda, yang digunakan untuk mengubah cycle. Pada cycle genap/setelah reset, Hasil read dari memory yang ditunjuk oleh Program Counter langsung masuk ke dalam bagian decoding, terletak di bawah kaca dan agak berantakan.



Informasi memory diubah menjadi binary untuk mempercepat transportasi



Bagian pink sambungan dari hasil read (kabel pink diatas untuk display). Kabel putih ditengah berarti akan dilakukan antara JUMP, ADD, atau SUB



Bagian belakang Control Unit yang terhubung dengan ALU. Kabel terdekat menandakan akan dilakukan operasi matematika (MATH MODE), kabel ujung menandakan operasi yang dilakukan (+ atau -).

HASIL



CPU versi 1, bisa melayani penambahan dan pengurangan
Untuk membuat program, harus langsung ditulis ke memori satu per satu.

Video demo : <https://youtu.be/8nxqX0eY6SE>

REFERENSI

Eksplorasi awal mengenai bagaimana CPU bekerja :

<https://www.youtube.com/watch?v=8ugmd6PBrPw>

<https://www.youtube.com/watch?v=4knBXkN1GEU>

https://youtu.be/7J7X7aZvMXQ?si=6zvd_yfWY5mBIVFi

Referensi gambaran umum yang menjadi inspirasi utama untuk desain ini :

<https://youtu.be/Z5JC9Ve1sfI?si=CYrbduGBeHIWwC7Z>

Kompartemen/circuit redstone :

https://www.reddit.com/r/redstone/comments/sq5vo9/hex_to_binary_v2/

<https://www.youtube.com/watch?v=-AxxhRtx1nA>