

Thomas Glansdorp
Ziggy Weesing
Zoe Davidson
ZtZ project

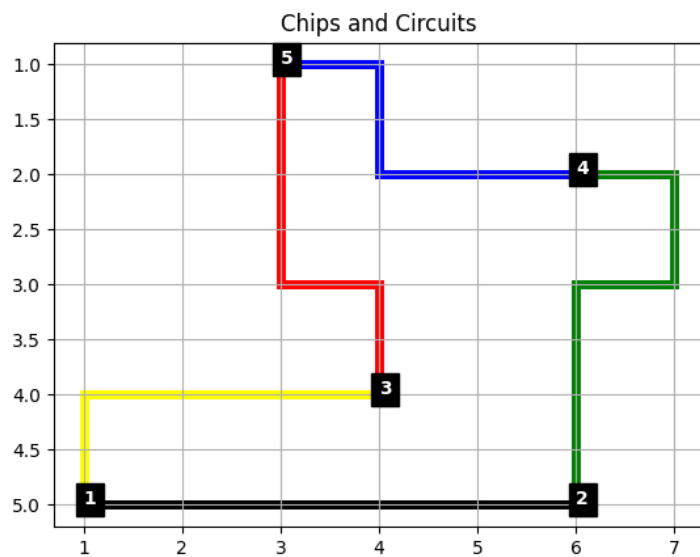
Baseline Chips & Circuits

Ons random algoritme werk zo dat er een pad gevonden wordt tussen twee gates door random richtingen op te gaan tot het pad bij de tweede gate is. Als het programma vastloopt, omdat er geen correct pad meer gevonden kan worden, zal het opnieuw runnen om toch een oplossing te vinden. Pas als er een geldige oplossing is gevonden laat hij dat ons zien. Dit hebben we allemaal gedaan met de netlist van de eerste chip. Met een netlist met meer connecties werkt het niet meer. Dit komt misschien omdat het voor die grotere netlist een 3D vak nodig is. Het kan ook komen omdat er hiervoor maar één oplossing is en dat ons random algoritme die niet kan vinden.

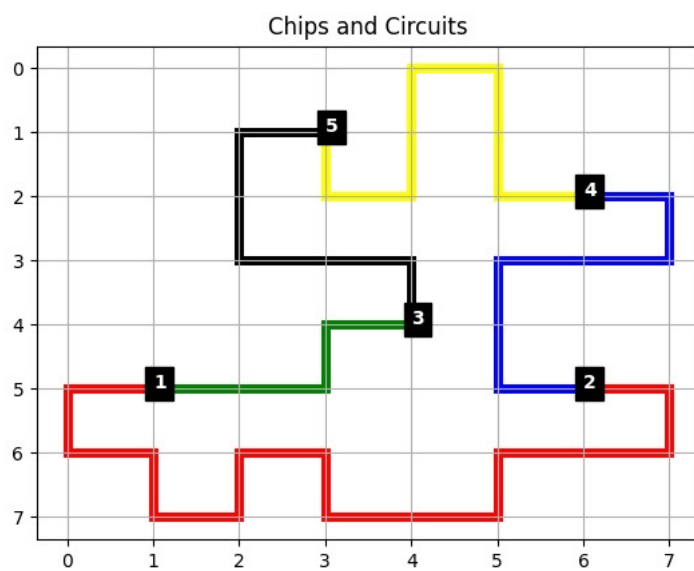
De resultaten die hieruit komen zijn eigenlijk altijd slechte oplossingen. Er komen bijna altijd oplossingen uit met een wirecount tussen de dertig en de vijftig. De beste oplossing heeft een wirecount van 20. Om tot een oplossing met een wirecount onder de 25 te komen, moet het programma tussen de 100.000 en 1.500.000 keer runnen. De paden zijn nog in een 2D vlak. We hebben hiervoor gekozen omdat het ons heel onhandig leek om een random algoritme te maken die in een 3D ruimte zijn pad zou gaan zoeken. Dit maakt het namelijk bijna onmogelijk dat er een oplossing uitkomt.

Nadat er een nieuw bord aangemaakt wordt om een nieuwe oplossing te zoeken, wordt de volgorde van de netlist geschud. Dit zorgt ervoor dat hij niet elke keer dezelfde volgorde aanhoudt bij het zoeken van de paden tussen de gates. Het viel ons namelijk op dat het eerste gekozen pad bijna altijd het langst was. Dit kwam natuurlijk omdat hij bij het eerste pad nog het hele bord heeft waarover hij kan lopen. Door de netlist te schudden, zorg je ervoor dat elke keer een ander pad het eerste pad is. Hierdoor haal je een deel van de bias weg.

Het is veel waarschijnlijker dat er een slechte oplossing met een hoge wirecount uitkomt dan een goede oplossing met een lage wirecount. Dit maakt de steekproef niet uniform. Dit is vrij logisch omdat de kans veel groter is dat hij in het pad een paar keer de verkeerde kant op gaat dan dat hij gelijk het juiste pad kiest. Met ons random algoritme kan je dus eigenlijk geen uniforme steekproef doen.



Figuur 1: wirecount 22



Figuur 2: wirecount 44