

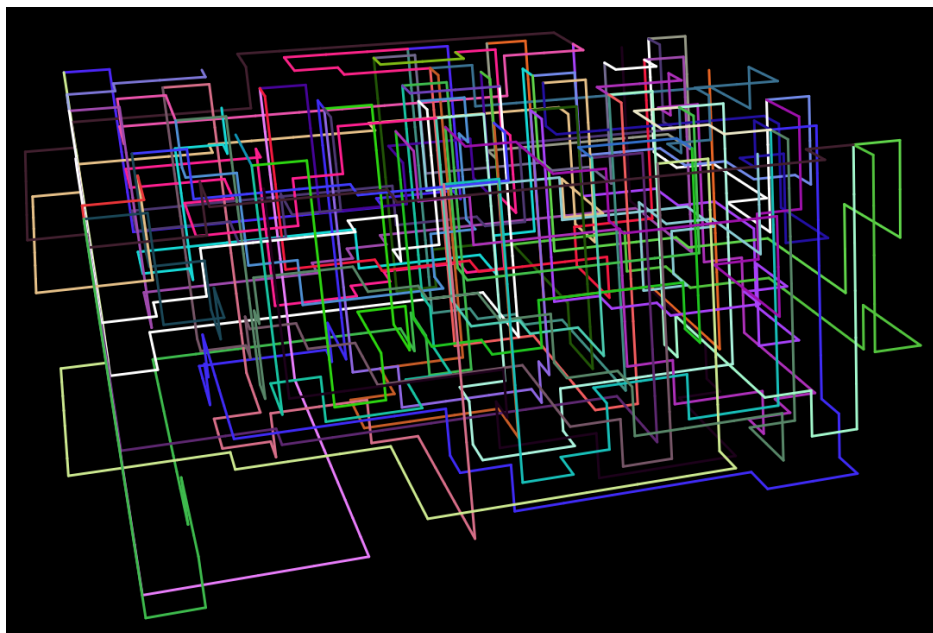
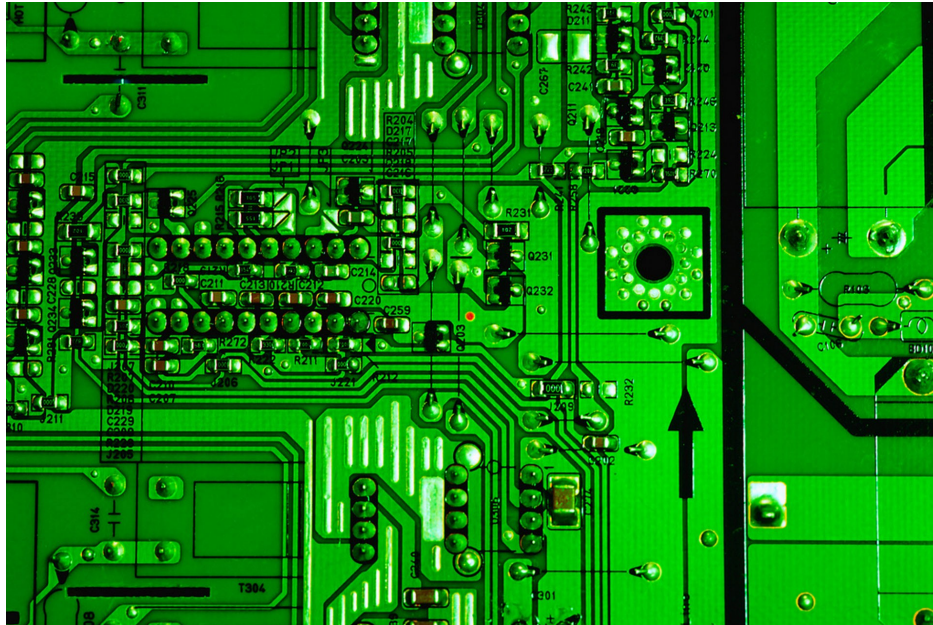
# Eindverslag ‘Chips & Circuits’

*Maurits de Ridder, Hasine Efeturk, Marianne de Heer Kloots*

Programmeertheorie, Groep D, ‘HaMaMa’

docenten: Daan van den Berg, Jelle van Assema, Maarten Inja

17-12-2015



# INHOUDSOPGAVE

<b>1. Inleiding</b>	<b>2</b>
<b>2. Methodes</b>	<b>2</b>
2.1 Vooraf: algoritmes . . . . .	2
2.1.1 BFS . . . . .	2
2.1.2 A* . . . . .	2
2.1.3 Netlists sorteren . . . . .	2
2.1.4 Kruisingen toestaan . . . . .	2
2.1.5 Loskoppelen gate-buren . . . . .	2
2.1.6 Toewijzing random layers . . . . .	2
2.2 Achteraf: optimalisatie . . . . .	2
Bla . . . . .	2
Bla . . . . .	3
<b>Resultaten en conclusie</b>	<b>3</b>

# 1. Inleiding

Elektronische chips (of *geïntegreerde circuits*) zijn te vinden in vrijwel elk elektronisch apparaat, van computers tot telefoons en magnetrons. Ze worden gebruikt om verschillende functies binnen deze apparaten uit te voeren, zoals het bijhouden van de tijd, maar ook bijvoorbeeld logische berekeningen. Computers kunnen met geen mogelijkheid zonder chips, en het optimaal configureren van deze chips is dus van essentieel belang. In dit verslag zullen we beschrijven hoe we de optimale configuratie van de circuits op een chip computationeel hebben proberen te simuleren, en de resultaten presenteren die daaruit zijn voortgekomen.

Een chip bestaat uit een klein siliconen plaatje met een aantal logische gates, die op een gespecificeerde manier met elkaar zijn verbonden door elektrische paden. Een chip kan bestaan uit meerdere lagen, waarbij alleen op de bovenste laag gates zitten; de elektrische paden kunnen dan over meerdere lagen lopen. Voor onze simulatie hebben we twee verschillende chipconfiguraties gebruikt, met verschillende groottes en hoeveelheden gates. Voor beide configuraties konden de gates op drie verschillende manieren verbonden worden. De zes lijsten van met elkaar te verbinden gates noemen we netlists. De netlists specificerden steeds een tiental van 30 tot 70 elektrische paden die gelegd moesten worden. De totale lengte van de paden moet zo kort mogelijk zijn, maar de te leggen paden mogen elkaar niet kruisen. Bovendien kunnen de paden uitwijken naar lagere lagen van de chip, maar in totaal kan de chip niet uit meer dan 8 lagen bestaan. ?? ZIE AFBEELDING ??

De toestandsruimte met alle mogelijke paden is enorm: voor alleen alle mogelijke kortste paden bij de kleinste configuratie in onze simulatie (een chip van 18 bij 13 met 25 gates) is bijvoorbeeld al  $1.763114 * 10^{105}$ . Het beschreven probleem is daarom een NP-hard probleem - dat wil zeggen dat het niet in polynomiale tijd is op te lossen door middel van een brute force techniek (het simpelweg uitproberen van alle mogelijke opties). Het gebruik van slimme algoritmes en goed gekozen heuristieken is daarom belangrijk om toch binnen een redelijke tijd een werkbare oplossing te kunnen vinden.

Voor onze simulatie hebben we verschillende zoekalgoritmen en heuristieken vergeleken. In het hoofdstuk Methodes zullen we die bespreken en vergelijken. In de hoofdstukken Resultaten en Conclusie zullen we vervolgens onze beste resultaten presenteren en advies geven voor verdere mogelijke optimalisaties van chipconfiguraties.  $\sum_{i=1}^{70} \frac{(|\Delta x[i]| + |\Delta y[i]|)!}{|\Delta x[i]|! * |\Delta y[i]|!} * 70!$

## 2. Methodes

### 2.1 Vooraf: algoritmes

#### 2.1.1 BFS

#### 2.1.2 A\*

#### 2.1.3 Netlists sorteren

Netlist ordenen kort-lang: 24,25,26; lang-kort 17-18

#### 2.1.4 Kruisingen toestaan

#### 2.1.5 Loskoppelen gate-buren

#### 2.1.6 Toewijzing random layers

### 2.2 Achteraf: optimalisatie

Bla

Bla bla

**Bla**

Bla bla bla

## **Resultaten en conclusie**

Bla!