

**MASTER**

**Compactie van bedradingskanalen in automatisch ontworpen IC's**

Huijnen, H.F.G.M.

*Award date:*  
1989

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Technische Universiteit Eindhoven  
Faculteit der Elektrotechniek  
Vakgroep Automatisch Systeemontwerpen (ES)

COMPACTIE VAN BEDRADINGSKANALEN  
IN AUTOMATISCH ONTWERPEN IC's

Verslag van een afstudeerproject  
uitgevoerd door H.F.G.M. Huijnen  
in opdracht van prof. dr. ing. J. A. G. Jess  
onder begeleiding van dr. ir. J. F. M. Theeuwen

aug. 1989

De Technische Universiteit Eindhoven is niet aansprakelijk  
voor de inhoud van stage- en afstudeerverslagen.

The Eindhoven University of Technology is not responsible  
for the contents of training and thesis reports.

## INHOUD

SAMENVATTING . . . . .	1
1. INLEIDING . . . . .	2
2. DE UITGANGSSITUATIE EN DE PROBLEEMSTELLING . . . . .	5
2.1 Inleiding . . . . .	5
2.2 De globale router . . . . .	7
2.2.1 Beschrijving . . . . .	7
2.2.2 De aansluitpinnen langs de omtrek van de layout . . . . .	8
2.3 De channelrouter . . . . .	9
2.3.1 Functie . . . . .	9
2.3.2 Technologie . . . . .	10
2.3.3 De werking . . . . .	11
2.3.4 Probleemstelling . . . . .	11
2.4 Mogelijkheden tot het verkleinen van het oppervlak . . . . .	11
3. AUTOMATISCH PLAATSEN VAN DE PINNEN AAN DE OMTREK VAN DE LAYOUT . . . . .	13
3.1 De vereiste gegevens . . . . .	13
3.2 Het merken van de netten als in- en uitgang . . . . .	13
3.3 Het bepalen van de layoutzijde waaraan een pin moet liggen . . . . .	14
3.4 Het vaststellen van de positie van de pinnen langs de zijde van de layout . . . . .	14
3.5 De communicatie met de gebruiker . . . . .	15
3.6 Resultaten . . . . .	15
3.7 Conclusies en aanbevelingen . . . . .	16
4. COMPACTIE - INLEIDING . . . . .	17
4.1 Randvoorwaarden . . . . .	17
4.2 Compactiemethodes . . . . .	17
5. DATASTRUCTUREN . . . . .	20
5.1 Data-representatie . . . . .	20
5.2 De datastructuur voor de bedrading . . . . .	20
5.3 De datastructuur voor de contouren . . . . .	21
6. COMPACTIE MET BEHULP VAN CONTOUREN . . . . .	23
7. STRETCHING . . . . .	25
7.1 De grote lijnen van het algoritme . . . . .	25
7.2 De basisprincipes van stretching . . . . .	25
7.3 De voorwaarden voor stretching . . . . .	26
7.4 Stretching: de details . . . . .	26
7.5 Resultaten . . . . .	28
8. VERDERE OPTIMALISATIES . . . . .	29
8.1 De aansluitingen van de verticale kanalen op de horizontale . . . . .	29
8.1.1 Inleiding . . . . .	29
8.1.2 Probleemstelling . . . . .	29

8.1.3 De oplossing . . . . .	29
8.2 De gegevensoverdracht . . . . .	30
8.3 Het compactie-algoritme op het laagste niveau . . . . .	31
9. RESULTATEN . . . . .	32
10. CONCLUSIES EN AANBEVELINGEN . . . . .	37
10.1 Conclusies en aanbevelingen met betrekking tot het automatisch plaatsen van de pinnen aan de omtrek van de layout . . . . .	37
10.2 Conclusies en aanbevelingen met betrekking tot compactie en stretching . . . . .	37
LITERATUUR VERWIJZINGEN . . . . .	38
APPENDIX . . . . .	39

## LIJST VAN FIGUREN

Figuur 1. Voorbeeld van een standaardcel . . . . .	3
Figuur 2. De delen van de logische synthese . . . . .	5
Figuur 3. Globaal overzicht van de layout . . . . .	6
Figuur 4. Aansluitpinnen langs de omtrek van de layout . . . . .	9
Figuur 5. Kanaal, sterk schematisch . . . . .	10
Figuur 6. Stretching . . . . .	27
Figuur 7. Genormeerde rekensnelheden tegen rekenduur . . . . .	36

## LIJST VAN TABELLEN

TABEL 1. Oppervlaktereductie . . . . .	33
TABEL 2. Rekensnelheid . . . . .	34
TABEL 3. Reproduceerbaarheid van de tijdmeting . . . . .	35

## SAMENVATTING

De bij de vakgroep "Automatisch Systeem Ontwerpen" van de Technische Universiteit Eindhoven ontwikkelde silicon compiler is uitgebreid om compactere resultaten te bereiken. De oppervlaktereductie wordt bereikt door compactie van de kanalen en door automatisch plaatsen van de aansluitpinnen aan de omtrek van de layoutmodules.

Bij de compactie wordt de contourmethode gebruikt, gevolgd door stretching. De resultaten zijn goed. Typische waarde voor de reductie van het behandelde oppervlak is 23%. Het programma dat de compactie thans voorafgaat, de router, vereist een vaste toewijzing van layoutlagen aan de bedrading. Het compactie-algoritme hoeft niet aangepast te worden wanneer deze restrictie vervalt en het laat zich met minimale veranderingen inzetten wanneer de technologie meerlaags bedrading toestaat.

Het automatisch plaatsen van de pinnen aan de omtrek van de layout verkleint het benodigde oppervlak en neemt de gebruiker een lastig werk, waarvoor hij niet altijd over voldoende gegevens beschikt, uit handen. De gebruiker kan kiezen uit volledig automatische plaatsing, de oude manier en enkele tussenvormen.

## 1. INLEIDING

Dit verslag behandelt een compactieprogramma dat deel uitmaakt van een "silicon compiler" die in de vakgroep "Automatisch Systeem Ontwerpen" van de Technische Universiteit Eindhoven is ontwikkeld. Een silicon compiler is een groep programma's. Te zamen zijn deze in staat om, uitgaande van een functionele beschrijving in een hogere taal zoals C, Pascal en dergelijke, een IC (chip) layout te genereren [1].

Het functioneren van een IC wordt vastgelegd door een aantal fotografische maskers die bij de fabricage worden gebruikt. Zo'n set maskers wordt een layout genoemd. De data die door de silicon compiler geproduceerd worden, vormen te zamen een volledige beschrijving van een maskerset, en worden dan ook wel aangeduid met "een layout".

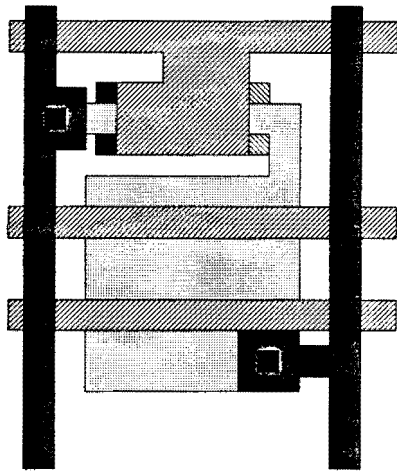
De silicon compiler bestaat uit een aantal grote delen. Slechts drie daarvan worden hier aangestipt:

- De "architectuur-synthese" genereert uit de functionele omschrijving op hoog-niveau grote blokken zoals datapad, bus, combinatoriek en dergelijke. Zij specificeert ook welke signalen ieder blok in- en uitgaan.
- De "floorplanner" verdeelt deze grote blokken over de oppervlakte. Hij kent ieder blok een plaats toe, en geeft aan waar bij elk blok de in- en uitgangen dienen te zitten.
- De "logische synthese". Voor elk soort blokken dat de architectuur-synthese maakt heeft de silicon compiler specifieke deelprogramma's die blokken van hun type moeten realiseren. De "logische synthese" is er daar één van. Zij concretiseert de combinatorische blokken. De architectuur-synthese levert als input voor de logische synthese een beschrijving die bestaat uit een stelsel booleaanse vergelijkingen die een of meer combinatorische functies specificeren.

Voor dit afstudeerproject is vooral de "logische synthese" van belang. Ze valt in enkele grote brokken uiteen (zie fig. 2). Allereerst worden de vergelijkingen die zijn ingevoerd door enkele programma's geanalyseerd en geoptimaliseerd. Daarbij is het doel de gegeven functie met een minimum aan cellen te realiseren. Cellen of standaardcellen zijn standaarddeelfuncties waarvan alle gewenste gegevens, zoals bijvoorbeeld de layout, voorhanden zijn.

Log\_sim en log\_decom herschrijven de gegeven functies zó dat ze een minimaal aantal termen bevatten. Log\_sim verwijderd redundantie uit de expressies; log\_decom onderzoekt of subexpressies op meer plaatsen voorkomen en substitueert daar nieuwe variabelen voor. Log\_net, thans vervangen door log\_mapper, vertaalt de bewerkte functies in een functionele beschrijving van de benodigde standaardcellen en een netlist waarin de verbindingen van deze cellen worden opgesomd. Het verkregen tussenresultaat wordt opgeslagen in de "gate-file".





Figuur 1. Voorbeeld van een standaardcel

De celgenerator, "log\_cel", genereert voor ieder van de cellen in de gate-file een deel-layout (zie fig. 1). Het uitgangspunt daarvoor is de functionele beschrijving van de cel, zoals die in de gate-file gevonden wordt. Er zij op gewezen dat - in tegenstelling tot wat de zeer geschematiseerde figuren misschien suggereren - de cellen afhankelijk van hun functie in hoogte kunnen variëren. Hun breedte daarentegen ligt vast. Alle data van de cellen, op hun layout na, worden in de database geplaatst.

De verdeling van de noodzakelijke cellen over het platte vlak is het volgende punt van aandacht. Men tracht dat zo te doen dat het totale in beslag genomen oppervlak minimaal wordt. Het programma dat deze taak verzorgt heet "log\_place". De gezamenlijke oppervlakte van de cellen is in dit stadium bekend; de keuzevrijheid ligt bij het oppervlak dat nodig is voor de verbindingen. Bij de plaatsing streeft men doorgaans naar minimale afstanden tussen de cellen die veel onderlinge verbindingen hebben om op die manier de totale lengte aan bedrading - en daarmee het vereiste oppervlak - klein te houden. De standaardcellen worden door de placer tot kolommen gestapeld (zie fig 3).

Log\_place betreft al zijn gegevens uit de database en zet er zijn resultaten in terug.

De laatste stap van de logische synthese is het maken van alle noodzakelijke verbindingen. Deze bewerking staat bekend als routing.

Een net wordt gedefiniëerd door alle punten in de layout die dezelfde potentiaal voeren. Voor ieder net verbindt de router alle ertoe behorende punten met elkaar. Het belangrijkste neven-doel van de router is het minimaliseren van het oppervlak dat de bedrading beslaat.

Dit verslag beschrijft twee methodes om bij de bestaande router te komen tot een verdere verkleining van het oppervlak.

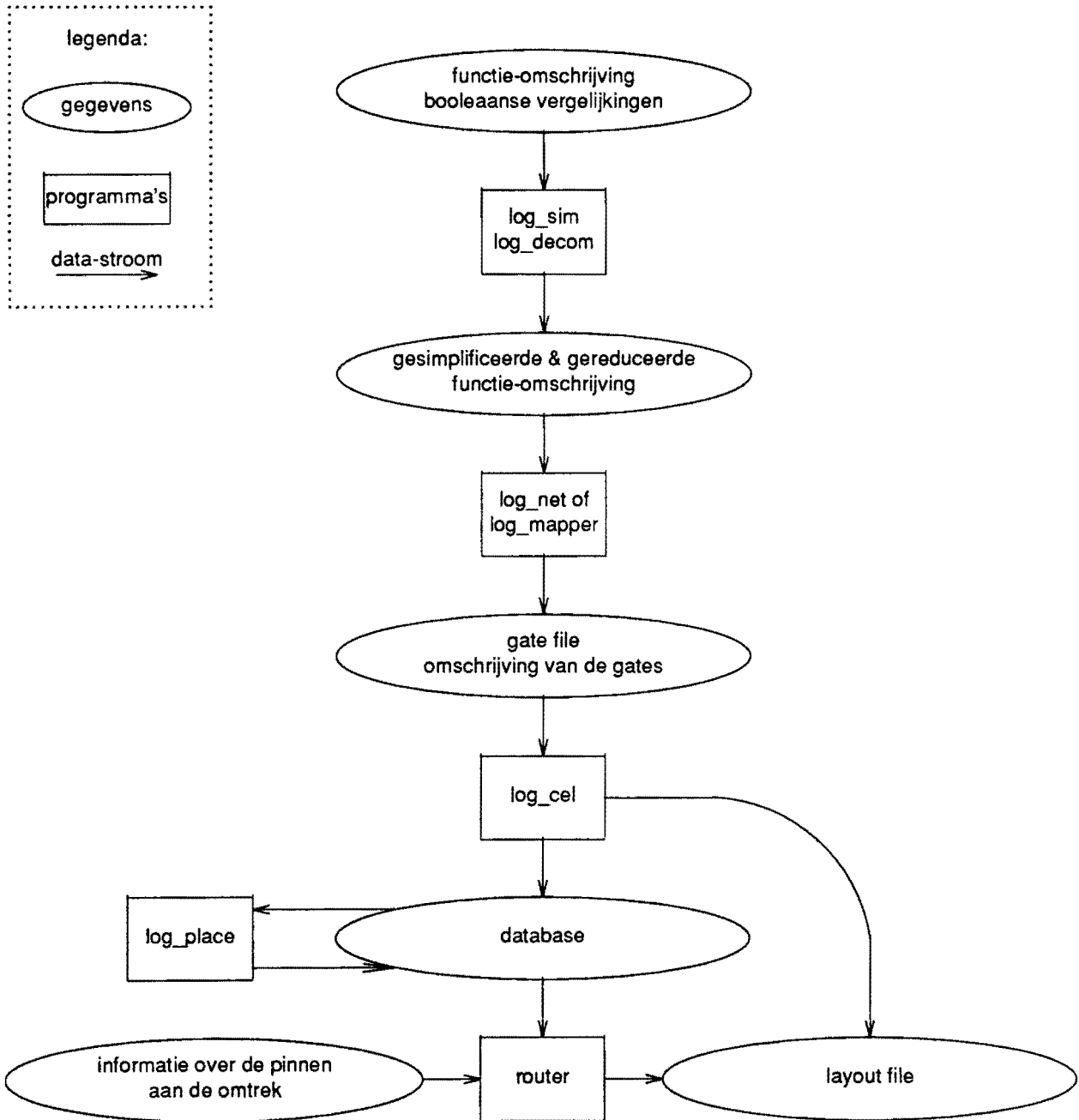
- *Automatisch plaatsen van de pinnen aan de omtrek van de layout*, die de verbindingen met de omgeving verzorgen (in- en uitgangen). Wanneer de silicon compiler de logische synthese start zijn in feite van alle in- en uitgangen de meest gewenste posities bekend. De uitwisseling van deze data tussen de silicon compiler en de laatste stap van de logische synthese, de router, is tot op heden niet geregeld. Daardoor moest de gebruiker voorheen voor elke pin coördinaten invoeren. Dat is een onaangenaam werk, waarbij nogal eens fouten gemaakt worden. Vaak ontbreekt het de gebruiker aan voldoende gegevens om voor iedere pin de juiste plaats te specificeren, wat resulteert in een onnodig grote layout. Er is een programma gemaakt dat de plaatsing van deze pinnen kan verzorgen. Daarbij heeft de gebruiker de keuze tussen volledig automatische plaatsing, de oude methode en enkele mengvormen. Wanneer in de data-uitwisseling is voorzien blijft het grootste gedeelte van het programma van nut.
- *Compactie*. De taak van de router is minder complex gemaakt door enkele aannamen. Het daaraan verbonden nadeel is dat de bedrading meer ruimte inneemt dan strikt genomen nodig is. Door de bedrading in de kanalen samen te persen wordt ruimte gewonnen. Dit proces heet compactie [2-8]. De omstandigheden waaronder de in de literatuur beschreven programma's werken verschilt in veel gevallen van de actuele situatie, waardoor de meeste methodes afvallen.

In hoofdstuk 2 wordt allereerst de aanvangssituatie in ogenschouw genomen. Daarbij valt de nadruk op de router. Uit deze bespreking volgt de probleemstelling. Er worden dan enkele mogelijke oplossingen besproken, waarvan er twee in de volgende hoofdstukken worden uitgewerkt. Zo wordt in hoofdstuk 3 het automatisch plaatsen van de pinnen voor de in- en uitgangssignalen besproken. In hoofdstuk 4 wordt een inleiding op de compactie gegeven. Daarbij blijkt dat er ná de compactie een extra bewerking, de stretching, nodig is. In de hoofdstukken 5, 6 en 7 wordt dieper ingegaan op de compactie en de stretching. Ze behandelen achtereenvolgens de datastructuren en de algoritmes. In hoofdstuk 8 komen enkele op zich staande onderwerpen aan bod, die zowel met de compactie als met de stretching te maken hebben. De resultaten worden gepresenteerd in hoofdstuk 9 en hoofdstuk 10 tenslotte bevat de conclusies en aanbevelingen.

## 2. DE UITGANGSSITUATIE EN DE PROBLEEMSTELLING

### 2.1 Inleiding

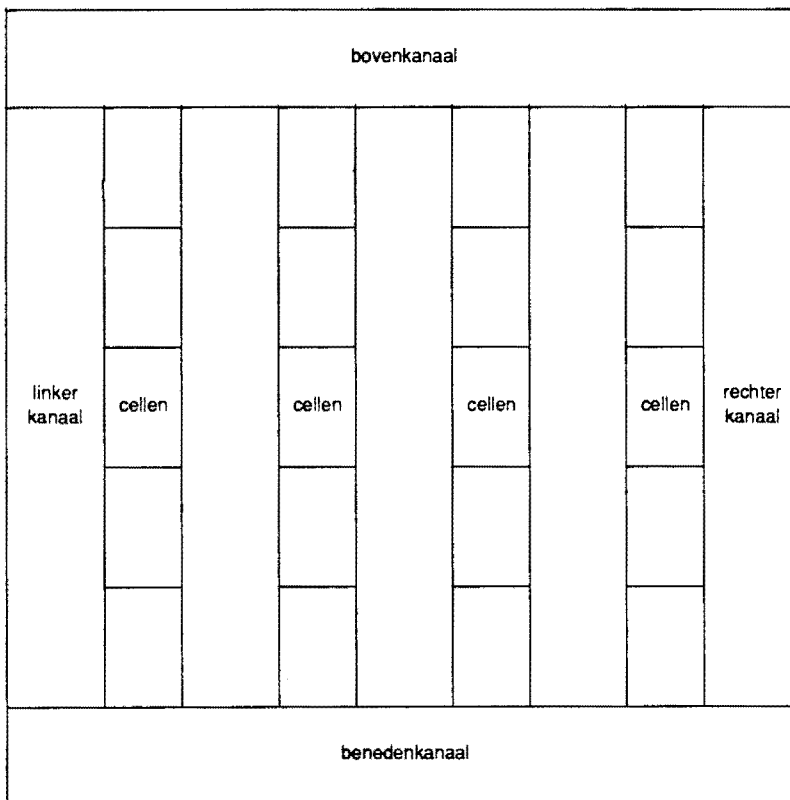
De logische synthese bestaat uit een keten van programma's (zie fig. 2) [9].



**Figuur 2.** De delen van de logische synthese

De programma's die de booleaanse vergelijkingen zodanig bewerken dat realisatie in een beschikbare technologie mogelijk is, zijn van essentieel belang voor een goed resultaat. Toch worden ze niet verder besproken, want ze zijn niet noodzakelijk voor een goed begrip van de hier gepresenteerde uitbreidingen. Om dezelfde reden worden de celgenerator en de placer niet nader toegelicht. In het vervolg van dit hoofdstuk wordt de router behandeld.

In dit verslag is een kolom - tenzij uit de context duidelijk anders blijkt - een kolom van cellen, zoals die door de placer wordt gemaakt. De rechthoekige ruimten die de kolommen in de layout uitsparen worden kanalen genoemd en bevatten de bedrading. De kanalen tussen de kolommen, het kanaal links van de kolommen en dat rechts ervan, heten de verticale kanalen. Het kanaal aan de boven- en dat aan de onderzijde vormen de horizontale kanalen. De vier kanalen die aan de omgeving grenzen, worden wel aangeduid als randkanalen (zie fig. 3).



Figuur 3. Globaal overzicht van de layout

## 2.2 De globale router

### 2.2.1 Beschrijving

De globale routing is de eerste van de twee stappen waarin de routing is verdeeld. Hierbij wordt beslist langs welke kanalen de netten die in meer dan een kanaal voorkomen, gevoerd worden. De leidraad voor de keuze is minimale totale draadlengte en daarmee impliciet minimale oppervlakte.

Wanneer twee cellen die niet aan hetzelfde kanaal grenzen, met elkaar moeten worden verbonden, kan de verbinding op verschillende manieren tot stand komen.

- Ze kan via het bovenkanaal geleid worden,
- of via het benedenkanaal.
- Bij kanalen die naast elkaar liggen bestaat de kans dat in de kolom cellen, die beide kanalen scheidt, een cel ligt die eveneens aan dit net moet worden aangesloten. De aansluitingen van standaardcellen zijn links/rechts-symmetrisch (zie fig. 1). Daardoor functioneren deze cellen ook als doorverbinding. Wordt een cel aangesloten op een net dat in het kanaal links van de cel aanwezig is, dan is dat net hierdoor ook in het kanaal rechts van de cel present.
- Is er in de tussenliggende kolom geen cel aan dit net aangesloten, dan is het mogelijk om op een geschikte hoogte in de kolom van cellen een "feedthrough" op te nemen. Een feedthrough is een zeer platte cel, even breed als de standaardcellen, die slechts bestaat uit een verbinding van links naar rechts. Zo kan er een net van het ene kanaal naar het naastliggende gebracht worden, zonder dat daarvoor steeds het boven- of benedenkanaal gebruikt wordt.

Het streven bij deze beslissingen is de totale draadlengte - en daarmee de noodzakelijke oppervlakte - zo klein mogelijk te houden.

Samenvattend:

- De globale router beslist welke netten door welke kanalen zullen lopen.
- Hij maakt er gebruik van dat de standaardcellen een doorverbinding vormen, waardoor een net in beide naast de cel liggende kanalen beschikbaar is.
- Zo nodig voegt de globale router een feedthrough toe.
- Het doel waar de globale router naar streeft is minimale draadlengte.

Wanneer de globale router zijn taak beëindigd heeft, zijn de randen van alle kanalen vastgelegd. Dat wil zeggen: voor ieder kanaal is bekend op welke rand, en op welke positie langs die rand een net geïntroduceerd wordt. Zo'n aansluitpunt van een cel in een kanaal of van het ene kanaal in het andere heet pin.

De globale router heeft twee bronnen van informatie. Op de eerste plaats is er de database-file. Zij bevat informatie over de coördinaten van de aansluitpunten van de netten aan de linker- en rechterzijde van elke kolom cellen. De tweede informatiebron moet gegevens leveren over de externe aansluitpinnen. Het is de architectuur-synthese die deze data genereert, maar er is nog geen bedding voor deze gegevensstroom gerealiseerd. Tot zo lang dient de gebruiker deze gegevens te leveren. Voor ieder net in de layout wordt hem gevraagd of dat net een aansluitpin naar de omgeving moet krijgen en zo ja, aan welke zijde van de layout en op welke coördinaat langs die zijde de aansluitpin geplaatst moet worden. Het vergt weinig fantasie om te beseffen dat dit een vervelende taak is, en bovendien mag men verwachten dat daarbij vergissingen worden gemaakt. Belangrijker nog is dat de gebruiker in veel gevallen niet genoeg gegevens heeft om een optimale keuze te doen.

### 2.2.2 De aansluitpinnen langs de omtrek van de layout

Dit onderwerp is een nadere beschouwing waard. De gevolgen van slecht geplaatste pinnen langs de rand laten zich op twee niveaus voelen.

- *Lokaal, in het randkanaal.* Wordt een aansluitpin aan de rand van de layout in horizontale richting\* onnodig ver verwijderd van het punt waar het betreffende net vanuit de layout het randkanaal binnenkomt, dan is een horizontale draad nodig om beide te verbinden. Voor zover deze draad zich uitstrekt langs een druk bezet deel van het kanaal moet er extra ruimte worden gemaakt. Verticale draaddelen echter dragen niet bij aan de hoogte van het kanaal. De kortste horizontale lengte (0) wordt bereikt wanneer het mogelijk is de aansluitpinnen tegenover elkaar te zetten.
- *Globaal, in de layout.* Het plaatsen van een aansluitpin aan de ongunstige zijde van de layout betekent dat er in de layout aanzienlijk meer ruimte wordt ingenomen door de extra draad. Het oppervlak van de layout neemt toe, wat alleen maar acceptabel is, indien de toename gecompenseerd wordt met winst buiten de behandelde layout. Op dit moment zijn er over de omgeving geen gegevens beschikbaar en daardoor kan er hiervan geen afweging gemaakt worden.

Het komt niet zelden voor dat de layout deel is van een groter geheel. Soms worden de deellayouts dan naadloos tegen elkaar geplaatst (abutting). Pinnen moeten in dat geval precies op voorgeschreven posities langs de rand van de deellayout liggen, anders zijn de vereiste verbindingen niet gewaarborgd.

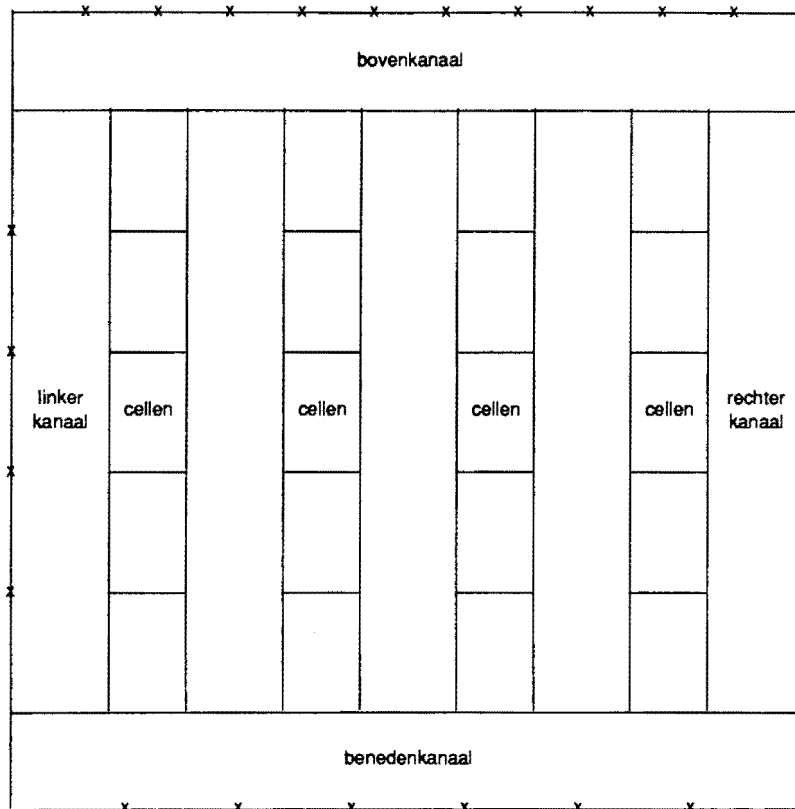
Doorgaans echter worden de deellayouts niet naadloos tegen elkaar geplaatst, maar gescheiden door een extern kanaal. De routing van naast elkaar liggende kanalen kost meer oppervlak dan wanneer deze kanalen gerout worden na vereniging. Er zijn redenen om desondanks de randkanalen niet in de externe kanalen te laten opgaan. De omgeving hoort niet tot de layout die in behandeling is en gegevens over de omgeving staan niet ter beschikking. In de randkanalen ligt ook bedrading die interne verbindingen van deze layout verzorgt. Worden de randkanalen afgestoten naar de omgeving dan mist men informatie over die interne verbindingen. De layout vormt geen geheel meer en het is niet meer mogelijk de layout te verifiëren.\*\* In deze situatie dienen de randkanalen zo slank mogelijk gemaakt te worden: alleen de verbindingen die intern nodig zijn horen erin gelegd worden. De verbindingen naar de omgeving moeten de randkanalen eenvoudig oversteken. Anders is de kans groot dat ze in het randkanaal een eind een richting uit geleid worden, om in het externe kanaal de andere kant uit gebracht te worden.

Ook wanneer de layout een volledig IC vormt heeft de gebruiker doorgaans onvoldoende gegevens om de aansluitpinnen goed te plaatsen. Van de layout staan de afmetingen op dat moment nog niet ter beschikking. Slechts het aantal kolommen van cellen en hun hoogte zijn bekend. De bedrading, welke in de komende stappen wordt toegevoegd, maakt doorgaans (veel) meer dan de helft van het layout oppervlak uit.

---

\* Alle kanalen worden bij routing behandeld alsof ze horizontaal zijn; de verticale kanalen worden, vóór plaatsing in de totale layout, 90° naar links gedraaid (tegen de wijzers van de klok in). Men denke zich als horizontale zijde steeds die zijde die aan een kolom cellen grenst, of waaraan twee of meer andere kanalen zijn aangesloten.

\*\* Het verificatieprogramma is het omgekeerde van de logische synthese. Uitgaande van een layout wordt een elektrisch schema van de schakeling gemaakt. Vervolgens wordt uit het schema gedestilleerd welke booleaanse functie er gerealiseerd is. Uiteraard hoort die equivalent te zijn aan de oorspronkelijke.



**Figuur 4.** Aansluitpinnen langs de omtrek van de layout

Coördinaten kunnen dus slechts ruw geschat worden. De volgorde van de aansluitpinnen is dan ook gemakkelijker op te geven dan de coördinaten. In een aantal gevallen zal de gebruiker slechts willen kunnen regelen aan welke zijde een aansluitpin komt te liggen en vaak wenst men gewoon een goede verdeling over de vier zijden.

Concluderend:

- Het mag niet nodig zijn dat de signalen die, als in- of uitgangssignaal dienen, als zodanig door de gebruiker gemerkt moeten worden.
- De tussenkomst van de gebruiker bij het vaststellen van de posities van pinnen langs de rand van de layout is ongewenst. De posities moeten automatisch worden gegenereerd, waarbij de gegevens die de omgeving betreffen (over bijvoorbeeld gewenste layoutzijde, volgorde van de pinnen langs een zijde) uit een file of database worden gelezen.

## 2.3 De channelrouter

### 2.3.1 Functie

Het vervolg op de globale router is de channelrouter, ook wel lokale router of kortweg router genoemd [10]. De router werkt maar in een kanaal tegelijk.

Voor elk net dat in het kanaal voorkomt verbindt de router de pinnen van dat net, die langs de randen van het kanaal liggen, zodanig dat de oppervlakte van het resulterende kanaal minimaal is en zo, dat voldaan wordt aan alle design rules (zie paragraaf 2.3.2).

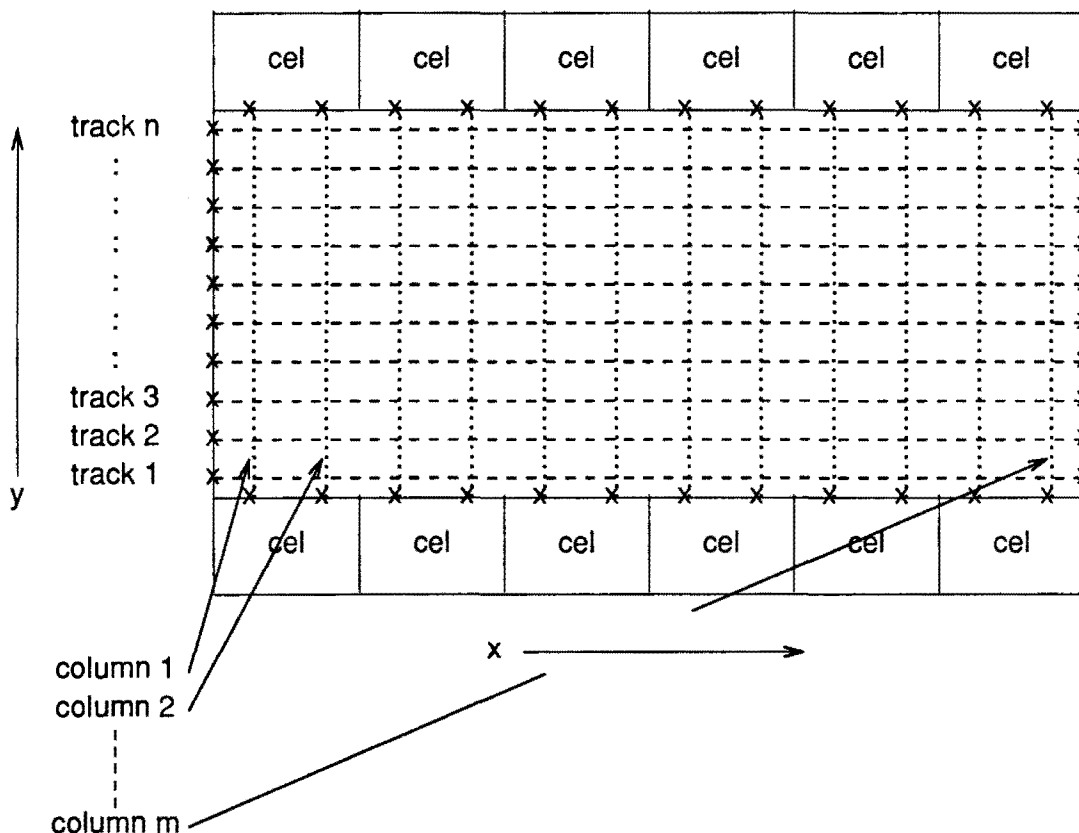
### 2.3.2 Technologie

Onder design rules verstaan we een set van regels waaraan een layout dient te voldoen om een goed werkend IC op te leveren. Gedacht moet worden aan zaken als minimale breedte voor ieder type layoutelement, minimale onderlinge afstanden, minimale overlap, maar ook gebods- en verbodsregels voor combinaties van layoutelementen. Layoutelementen kunnen eenvoudige rechthoeken zijn, maar ook complexe deellayouts, zolang ze als een geheel beschouwd worden.

Doorgaans beperkt men bij de IC fabricage de layouts tot vormen die uit rechthoeken kunnen worden samengesteld. Schuine lijnen, cirkels en dergelijke zijn niet toegestaan, wat een grote vereenvoudiging met zich mee brengt.

De opbouw van het IC gebeurt in lagen met elk hun eigen namen en eigenschappen. Verbindingen worden gemaakt in de lagen "interconnectie" (IN) en "polysilicium" (PS). Interconnectie bestaat uit aluminium en heeft een zeer lage weerstand. De vierkantsweerstand van polysilicium is circa 1000X zo hoog als die van de interconnectie. Aan draden wordt gerefereerd als liggend in een "laag" of "layer" en daar elke laag bepaald wordt door haar eigen masker, spreekt men ook wel van het "masker".

Interconnectie en polysilicium kruisen elkaar zonder elektrisch contact te maken. Elektrisch contact is mogelijk door het plaatsen van een "via" - een doorverbinding - op de kruising. Design rules vereisen dat de minimum afmetingen van een via groter zijn dan de minimum breedte van een draad, of die draad nu uitgevoerd is in interconnectie, of in polysilicium.



Figuur 5. Kanaal, sterk schematisch



### 2.3.3 De werking

De router gaat uit van een aantal premissen:

- *Orthogonale en equidistante bedrading.* Het kanaal wordt opgedeeld in "tracks" en "columns". Het zijn denkbeeldige lijnen, op gelijke afstand van elkaar, waarbij de tracks evenwijdig lopen aan de horizon en de columns daar loodrecht op staan. Horizontale draden liggen op de tracks, verticale op de columns. Met de horizontale richting wordt een x-coördinaat geassocieerd, met de verticale een y-coördinaat.
- *Een grid.* De tracks en columns vormen samen een rooster dat wordt aangeduid met "grid". Een kruispunt van een track en een column wordt "gridpunt" genoemd. De afstanden van track tot track en van column tot column zijn zo gekozen dat op ieder gridpunt een via geplaatst kan worden, ook als alle naaste gridpunten met een via bezet zijn.
- *De aansluitpunten van de cellen die aan het kanaal grenzen liggen op het grid.*
- *Vaste indeling van de draden in lagen.* De lange horizontale draden worden vanwege de lage elektrische weerstand uitgevoerd in interconnectie, de kortere verticale in polysilicium.

Alle kanalen worden bij routing behandeld alsof ze horizontaal zijn; de verticale kanalen worden, voor plaatsing in de totale layout, 90° naar links gedraaid (tegen de wijzers van de klok in).

De router gaat van links naar rechts door het kanaal. Hij maakt column na column gereed.

Het is een "greedy" router [11], waarmee bedoeld wordt dat de router op ieder moment de beslissing neemt die op dat moment de gunstigste lijkt, zonder in de toekomst te kijken. Bij deze router, die columns-gewijze werkt, houdt dat in dat zoveel mogelijk verbindingen gemaakt worden in de beschikbare hoogte van de column, zolang daar aanleiding toe is.

### 2.3.4 Probleemstelling

Het werk van de router wordt in hoge mate vereenvoudigd door de a priori aannamen. Ze leiden ertoe dat de router zich in een minder gecompliceerde omgeving beweegt, waardoor hij sneller en efficiënter kan functioneren, maar dat anderzijds nog veel te verbeteren is aan de wijze waarop van het IC oppervlak gebruik wordt gemaakt. De maaswijdte van het grid is afgestemd op het kunnen accepteren van via's op iedere kruising. Op veel plaatsen ligt er geen via, maar een draad, die veel smaller is. De gridlijnen zijn echter equidistant en daardoor is het verschil in breedte verloren ruimte.

De taak waarvoor we gesteld zijn is het reduceren van de hoogte van het kanaal met behoud van de door de router gemaakte verbindingen van de pinnen langs het kanaal en met inachtnaam van de design rules.

## 2.4 Mogelijkheden tot het verkleinen van het oppervlak

**Reduceren** van de grootte van de vier randkanalen door het automatisch plaatsen van de pinnen aan de rand van de layout. De winst zal sterk afhankelijk zijn van het aantal verbindingen dat de layout met de omgeving heeft. Bij kleine layouts met weinig kanalen is het vrijwel verdwijnen van de vier randkanalen verhoudingsgewijs een grote besparing. Bij grote layouts is de oppervlaktevermindering weliswaar in procenten uitgedrukt geringer maar door de grote lengte der randkanalen betekent elke krimp van de hoogte van een randkanaal toch een belangrijke verkleining van oppervlak.

**Compacteren** van de gerealiseerde bedrading. Dat kan slechts in één richting gebeuren, loodrecht op de horizontale as. De cellen langs het kanaal behouden immers hun afmetingen en compactie in de richting van de horizon zou de verbindingen met de pinnen verbreken.

Een belangrijk begrip is de "densiteit". Onder de "densiteit" verstaan we het aantal netten dat zowel links als rechts van een willekeurig gekozen verticale doorsnede van het kanaal tenminste één pin heeft.

Laten we de doorsnede langs het hele kanaal lopen, dan is de "maximale densiteit" de maximumwaarde die de densiteit aanneemt. Het spraakgebruik is slordig en vaak duidt men de maximale densiteit aan met "de densiteit".

Het is uit de definitie duidelijk dat het onmogelijk is het kanaal zo te bedraden dat het kanaal minder tracks bevat dan overeenkomt met de maximale densiteit.

Beschouwen we een kanaal ter plekke van de maximale densiteit. Neem aan dat er hier géén via's liggen en dat er ongehinderd gecompacteerd kan worden. We simplificeren de situatie verder door aan te nemen dat de verschillende isolatie-afstanden die de design rules kennen alle even groot zijn. De gridafstand is gelijk aan de afmetingen van een via plus eenmaal de isolatie-afstand. De originele kanaalhoogte bedroeg

(aantal tracks) X (via-afmeting + isolatie-afstand).

Na compactie wordt dat

(aantal tracks) X (horizontale draadbreedte + isolatie-afstand).

Wanneer bijvoorbeeld geldt:

horizontale draadbreedte = isolatie-afstand =  $\frac{1}{2}$  X via-afmeting

is duidelijk dat er maximaal 33% reductie van de kanaalhoogte mogelijk is.

**Colouring.** Onder "colouring" wordt verstaan het toewijzen van layers (maskers) aan de bedrading [12, 13].

De rigide toekenning van de layers interconnectie aan de horizontale en polysilicium aan de verticale draden was noodzakelijk voor een ongecompliceerde routing. Colouring - na routing - volgens een goed algoritme maakt het aantal onnodige overgangen van interconnectie naar polysilicium veel kleiner. Dat houdt het verdwijnen in van even zovele doorverbindingen van laag naar laag (via's), die hinderlijke obstakels vormen bij de compactie.

Polysilicium mag ongestraft onder een interconnectie draad lopen. Worden bij colouring twee horizontale draden in verschillende lagen uitgevoerd, dan kan de ene draad onder de andere verdwijnen. Colouring, gedaan ná de routing en vóór de compactie verlegt de bovengrens van de met compactie te behalen ruimtewinst uit de vorige paragraaf, omdat nu niet meer geldt dat alle horizontale draden onderling op isolatie-afstand moeten blijven.

Colouring is niet geïmplementeerd, maar de compactie is zodanig opgezet dat ze ook na colouring probleemloos zal werken.

### **3. AUTOMATISCH PLAATSEN VAN DE PINNEN AAN DE OMTREK VAN DE LAYOUT**

#### **3.1 De vereiste gegevens**

Om op juiste wijze de pinnen langs de rand van de deellayout te plaatsen zijn er gegevens noodzakelijk. Met name twee vragen dienen er beantwoord te worden:

1. Welke netten moeten als in- of uitgang worden aangemerkt.
2. Aan welke layoutzijde moet de pin van het in- of uitgangnet worden geplaatst.

En eventueel ook nog:

3. Welke volgorde of coördinaten moeten de pinnen hebben die alle aan dezelfde zijde van de layout liggen.

De antwoorden op deze vragen zijn wel bekend, maar de beschikbaarheid van de gegevens is niet vanzelfsprekend. Laten we de situatie per vraag onder de loep nemen.

Ad 1. De architectuur-synthese bepaalt welke signalen de combinatorische blokken in- en uitgaan. In de files met booleaanse vergelijkingen die door haar geproduceerd worden zijn regels opgenomen die beginnen met het teken "#" gevolgd door de woorden "inputvars" of "outputvars" en die vervolgens de namen bevatten van de netten die als ingang, respectievelijk uitgang, zijn te beschouwen.

Ad 2. Het is de floorplanner die voorschrijft aan welke zijde van de deellayout de diverse pinnen moeten liggen. Helaas is er niet gespecificeerd waar en op welke wijze deze gegevens te vinden zijn. Het maakt weinig verschil of de gegevens niet bereikbaar zijn of dat ze helemaal afwezig zijn: er kan geen gebruik van gemaakt worden.

Ad 3. Hierbij geldt, mutatis mutandis, hetzelfde als bij punt 2. Indien de soort layout die gegenereerd wordt zich ertoe leent om cellen rechtstreeks tegen elkaar te plaatsen ("abutting") zal de floorplanner coördinaten afleveren, in andere gevallen een volgorde van de pinnen.

De oorzaak van deze problematische data-uitwisseling is gelegen in de staat van diverse samenstellende programma's van de silicon compiler. Vele zijn nog in ontwikkeling en lang niet alle communicatieprotocollen zijn uitgekristaliseerd. Het spreekt voor zich dat dit een vrij onbevredigende situatie is: de geproduceerde layouts vallen vaak onnodig groot uit en daarnaast is het voor de gebruiker heel onaangenaam om te moeten dienen als vervangende informatiebron. Vandaar dat er aan een - ten dele voorlopige - oplossing is gewerkt.

#### **3.2 Het merken van de netten als In- en uitgang**

Om de netten te vinden die als in- of uitgang moeten worden gemerkt is er aan de global router een parser toegevoegd. Deze onderzoekt een file op regels die beginnen met "#inputvars" of "#outputvars". Daarbij worden spaties of tabs ter weerszijde van het teken "#" genegeerd. Alle volgende woorden op deze regels worden beschouwd als namen van netten die als in- of uitgang dienen. Het aantal regels dat net-namen bevat is onbeperkt en de regels mogen ook in iedere willekeurige volgorde voorkomen.

De onderzochte file is een andere dan de database. Het moet een bestaande ascii-file zijn die desnoods leeg mag zijn. Dit format wordt gehanteerd in de file met functieomschrijvingen en ook in de gate-file. Een van beide files kan gekozen worden.

Is de file leeg of ontbreken beide herkenningswoorden, dan wordt net voor net aan de gebruiker gepresenteerd met de vraag of dit net een "I/O-pin" (in- of uitgang) moet hebben. Zo is verzekerd dat de juiste netten een aansluitpin krijgen.

### 3.3 Het bepalen van de layoutzijde waaraan een pin moet liggen

Informatie hierover is wel aanwezig maar, door het ontbreken van afspraken over de gegevensuitwisseling, niet beschikbaar. Om die reden wordt er uitsluitend uitgegaan van gegevens die in deze deellayout te vinden zijn.

Het toegevoegde programmadeel dat een layoutzijde toekent aan elk in- of uitgangnet is een van de eerste procedures die in de globale router worden aangeroepen. Een net dat in het linker- of rechterkanaal voorkomt, krijgt een aansluitpin in dat kanaal. Van netten in de overige verticale kanalen wordt onderzocht op welk punt ze het verst verwijderd zijn van het geschatte midden van de hoogte van de hoogste kolom cellen. Ligt dat extreme boven het geschatte midden, dan wordt de pin aan de bovenzijde van de layout geplaatst; in het andere geval aan de benedenzijde.

Zodra een communicatieprotocol vastligt, dient het programmadeel dat een layoutzijde toekent vervangen te worden door een parser. Procedures daarvoor zijn beschikbaar; de parser die de netten als in- of uitgang merkt, is geheel uit algemeen bruikbare procedures opgebouwd.

### 3.4 Het vaststellen van de positie van de pinnen langs de zijde van de layout

Evenals de vorige gegevens zouden deze data met behulp van een parser uit de juiste file verzameld moeten worden. Nu de gegevensuitwisseling niet is vastgelegd rest er weinig anders dan de gebruiker de gegevens te laten leveren. Daarbij heeft deze drie mogelijkheden, die voor elke layoutzijde apart gekozen kunnen worden:

- *Automatisch.* De lengte van de layoutzijde wordt geschat. Voor elke pin wordt een coördinaat gegenereerd die op grid ligt. De eerste wordt zo dicht mogelijk bij het geschatte midden van de zijde gelegd, de volgende om beurten op gridposities erboven en eronder. Het doel van deze opzet is te voorkomen dat de global router één of twee kanalen overbelast.  
Verder wordt de layoutzijde gemerkt met "free" (vrij verschuifbare pinnen).
- *Op volgnummer.* De gebruiker dient elke pin een volgnummer te geven. De volgnummers worden meteen in vaste coördinaten, corresponderend met het grid, vertaald. De zijde wordt gemerkt met "restr" (schuiven van de pinnen onder restricties).
- *Met absolute coördinaten.* De gebruiker kent elke pin een coördinaat toe. De ingegeven waarde wordt naar het dichtstbijzijnde gridpunt afgerond. De zijde wordt gemerkt met "abs" (schuiven van de pinnen niet toegestaan).

De global router gebruikt de resulterende coördinaten om de draden op de juiste wijze door de diverse kanalen te routen.

Wanneer het om een randkanaal gaat, worden, onmiddellijk vóórdat de channelrouter het kanaal gaat routen, opnieuw coördinaten toegekend aan de pinnen die als in- of uitgang functioneren.

Bepalend is daarbij het label dat de layoutzijde draagt:

- "free" Voor iedere pin wordt gekeken of het net waar de pin deel van uitmaakt, aan de tegenoverliggende horizontale zijde aanwezig is. Zo ja, dan krijgt de pin dezelfde x-coördinaat zodat beide aansluitpunten recht tegenover elkaar komen te liggen. Zo niet, dan wordt gekeken of het net zijn entree maakt links of rechts in het kanaal en wordt de pin zo dicht mogelijk aan de zijde van binnenkomst geplaatst.
- "restr" Met behulp van een eenvoudige heuristiek wordt getracht elke pin een x-coördinaat te geven binnen het gebied waarin het net van die pin al in het kanaal aanwezig is, of tenminste daar zo dicht mogelijk bij. Daarbij wordt de volgorde van de pinnen langs de rand behouden, vandaar "restricted".
- "abs" De coördinaten worden niet veranderd.

### 3.5 De communicatie met de gebruiker

De communicatie met de gebruiker is gebaseerd op enkele overwegingen:

Invoer van de gebruiker die op eenvoudige wijze als foutief te herkennen is, mag niet de oorzaak zijn van voortijdig afbreken van het programma. Lukt het bijvoorbeeld niet om een antwoord als integer te interpreteren of ligt het antwoord buiten het gebied dat is toegestaan, dan wordt het verworpen. Datzelfde gebeurt bij respons die uit een of meer letters bestaat, wanneer die niet binnen de verzameling toegestane antwoorden ligt. De vraag wordt zolang herhaald, totdat de reactie niet meer als foutief wordt herkend.

Wanneer de gebruiker een lege regel als invoer geeft, moet een zo gunstig mogelijk standaardantwoord gesubstitueerd worden.

Lezen van een lege file in plaats van lezen van het toetsenbord moet hetzelfde zijn als een reeks van lege antwoorden; op deze manier moet het eenvoudig mogelijk zijn om het programma 's nachts of op de achtergrond te laten werken.

### 3.6 Resultaten

Blad 2 van de appendix toont het resultaat van het originele route-programma. Het voorbeeld is  $1098 \times 930 \mu\text{m}$  groot en heeft een oppervlak van  $1.02 \text{ mm}^2$ . Worden de pinnen automatisch geplaatst, dan resulteert "auto-pin": te zien op blad 3 van de appendix en  $764 \times 696 \mu\text{m}$  groot. Het oppervlak is nu  $0.53 \text{ mm}^2$ , wat een afname van 48% betekent. De winst is even groot als geflatteerd. Het voordeel voor de behandelde deellayout is immers niet afgewogen tegen de kosten die deze lokaal gunstige plaatsing voor het geheel met zich meebrengt. Onder de huidige omstandigheden is het echter beter om de deellayout te reduceren zonder het geheel te bekijken dan om niets te optimaliseren. Wanneer de gegevensuitwisseling tot stand is gekomen, kan de plaatsing van de pinnen een gunstig resultaat opleveren onder de randcondities die architectuur-synthese en floorplanner stellen.

### **3.7 Conclusies en aanbevelingen**

Het is dringend noodzakelijk dat er voor de gegevens die architectuur-synthese en floor-planner produceren - desnoods tijdelijke - afspraken worden gemaakt omtrent de plaats waar en het format waarin ze kunnen worden aangetroffen.

De toegevoegde software is in staat alle vereiste functies te vervullen. Is de gegevens-uitwisseling geregeld, dan dient alleen het interactieve deel door een parser te worden vervangen.

Wanneer de coördinaten van de pinnen automatisch gegenereerd worden of door het opgeven van een volgorde van de pinnen tot stand komen, is het vermoedelijk gunstig de pinnen niet slechts in het midden van het kanaal te plaatsen, maar ze zo gelijkmatig mogelijk te spreiden langs de layoutzijde: de verdeling van de bedrading over de kanalen zal erdoor verbeteren.

## 4. COMPACTIE - INLEIDING

### 4.1 Randvoorwaarden

Er worden aan de compactie voorwaarden gesteld:

- Langs de boven- en onderrand van het kanaal liggen cellen die niet gecompacteerd worden. Wil men het contact met de pinnen van deze cellen niet verloren laten gaan, dan moeten de kanaalafmetingen in deze richting (x-richting) hetzelfde blijven.
- Draden die de verticale kanalen verlaten naar het boven of beneden kanaal, moeten correct aansluiten op het grid van dat horizontale kanaal. De routing van de horizontale kanalen is niet mogelijk wanneer daar niet aan is voldaan.
- Het aantal bochten dat de compactie introduceert, dient zo laag mogelijk te blijven. De extra draadlengte, die het meanderen met zich meebrengt, betekent een toename van de weerstand en de capaciteit van de draad, wat in het bijzonder bij polysilicium zeer nadelig is. Bovendien hebben bochten een negatieve invloed op de te verwachten levensduur van het IC.
- Er mogen door de compactie geen via's bijgemaakt worden. Via's zijn volumineus en er bestaat altijd een - zij het kleine - kans op slecht contact o.a. door metaalmigratie.
- De werking van de compactie mag niet afhankelijk zijn van de laag waarin de diverse draden liggen. Anders zou colouring vóór compactie niet meer mogelijk zijn.

### 4.2 Compactiemethodes

Compactie staat op dit moment sterk in de belangstelling [2-8, 14]. Er zijn dan ook diverse succesvolle manieren gepubliceerd. We noemen de belangrijkste:

- *De shear-line-methode.* Het aanschuiven gaat overal over dezelfde afstand, evenwijdig aan een van de "shear-lines". Er wordt een kolom van vrije ruimte gezocht, die aan een zijde van de layout begint en aan de tegenover liggende zijde eindigt. Lukt het niet om zo'n kolom te vinden, dan zoekt men naar dekolommen, zó dat de projectie van alle dekolommen een ononderbroken rechte vormt. De denkbeeldige verbindinglijn tussen het einde van de ene dekolom en het begin van de volgende staat loodrecht op de dekolommen en heet "shear-line". De omgeving van elke shear-line vraagt enige zorg, omdat de layoutelementen ter weerszijde ervan langs elkaar zullen schuiven bij compactie. Het zoeken naar de lege kolommen is [4] van orde  $N^2$ . De bezwaren tegen deze benadering richten zich op de uniforme verschuiving over de volle breedte van de layout en op de orde van het algoritme.
- *De virtueel grid-methode.* Weste [6] ziet deze methode als deel van een interactief compactieproces. Men legt de delen van de layout in cellen van een tweedimensionale matrix. De hartlijnen van de matrix componenten worden opgevat als een grid. De opzet is afwisselende compactie in de x- en in de y-richting. Werken in de x-richting gaat kolomsgewijze, in de y-richting per rij. We gaan uit van compactie in de x-richting. Noem drie opeenvolgende kolommen A, B en C. Van kolom A wordt cel voor cel de inhoud vergeleken met de overeenkomstige cel uit kolom B. Bevindt zich in de cel van kolom A een draad die ook wordt aangetroffen in de cel van kolom B en die zich voortzet in kolom C, dan wordt deze cel van kolom B als leeg beschouwd. Zijn beide cellen niet leeg, dan wordt met behulp van de design rules de minimaal vereiste afstand tussen beide cellen berekend. De maximum waarde van alle minimum afstanden tussen de cellen van de kolommen A en B wordt nu de afstand tussen de gridlijnen van kolom A en B. Het algoritme kent verfijningen, bijvoorbeeld

voor design rules die zich over meer dan de breedte van één kolom of rij uitstrekken en uitzonderingen zoals de behandeling van transistoren die standaard over drie matrix cellen verdeeld zijn.

Het algoritme is orde  $N$ . (" $N$ " is het aantal elementen dat gecompacteerd wordt.)

Ook deze compactor werkt over de volle breedte of hoogte van de layout, wat bij bewerking van de resultaten van de router niet adequaat is.

- *De "constraint-graph"-methode.* Er wordt een graaf gemaakt waarin de relatieve afstanden tussen de elementen en de vereiste minimale afstanden zijn opgenomen. Vervolgens wordt met de "langste-pad" benadering een oplossing gezocht.

Het zoeken naar de afstanden tussen de elementen is van orde  $N \log N$ , het construeren en het oplossen van de graaf zijn beide orde  $N$ . De meeste layoutelementen hebben weinig te maken met elementen die verder weg liggen, waardoor in de praktijk de orde tendeert naar  $N^{1.2}$ .

Deze methode heeft als voordeel dat ze haar inspanningen richt op die gebieden waar de knelpunten liggen. Het is tegelijk haar kracht en haar zwakte dat ze de problemen opspoot, maar niet oplost; daarvoor is een apart programmadeel nodig. Het is haar kracht omdat ze daardoor aan te passen is aan de omstandigheden waaronder ze wordt ingezet en ook omdat daardoor diepgaande ingrepen te realiseren zijn. Het is een zwakte omdat er geen garantie is dat het toegevoegde programma van een voldoende lage orde is. Wanneer voor drastische maatregelen gekozen wordt kan het nodig zijn na elke ingreep de graaf grondig te herzien, wat tijdrovend kan zijn.

De methode was een geschikte kandidaat; desondanks is de keuze op een andere gevallen.

- *De contourmethode.*

Eén van de horizontale randen van het kanaal wordt afgebeeld op een aantal contouren: voor elke bedradingslaag één contour. Een contour is een verzameling lijnstukken waarvan de projectie op de horizon een ononderbroken rechte zonder overlappingsen vormt. Ze vormt de grens van het gebied waarbinnen de bedrading gelegd kan worden. De bedrading uit het track, dat naast de contouren ligt, wordt aangeschoven in de richting van de contouren, zó dat de bedrading op precies voldoende afstand van de contouren blijft. De contouren worden nu aangevuld met informatie over de aangeschoven bedrading. Zo worden alle tracks op hun beurt behandeld.

Aangezien elke rij bedrading (track) haar voorgangster op de voet volgt, zijn in elk nieuw track de onregelmatigheden van alle voorgaande tracks zichtbaar. Er worden bijgevolg ontoelaatbaar veel bochten in de bedrading geïntroduceerd.

Er is daarom een nabewerking, de stretching, nodig die alle overbodige bochten verwijdert, zonder meer ruimte op te eisen. Ook de stretching werkt met contouren. Per track wordt er gekeken welk gebied de bedrading van dit track ter beschikking staat, en in die ruimte worden de draden zo recht mogelijk neergelegd.



Compactie met behulp van contouren is gekozen omdat er een aantal voordelen aan zijn verbonden:

- De orde is  $N$  waarin  $N$  het aantal aan te schuiven draadstukken is.
- De aard van het algoritme kent geen eindeloze loops, veroorzaakt door "cyclic constraints", die men met uitgekookte beslissingsstrategieën te lijf moet gaan.
- Kanaalranden die niet recht zijn vormen geen enkel probleem. Standaardcellen hoeven niet meer alle van dezelfde breedte te zijn, alleen de  $x$ -coördinaat van de pinnen dient op grid te liggen.
- Er is geen aanpassing nodig wanneer men colouring wil toepassen.
- Gebruik van meerlaags bedrading vergt slechts minimale aanpassingen.

Dat de orde van het algoritme zo belangrijk is komt door het grote aantal te compacteren layoutelementen. Er is geteld hoeveel draadstukken bij de compactie betrokken waren: dat varieerde van 32 bij een layout met 7 cellen tot 7118 bij 82 cellen.

Het is opvallend hoe weinig al deze algoritmen verschillen in de ingrepen die ze in de layout zelve doen: aanschuiven van de elementen en waar nodig het maken van "jogs" (een opeenvolging van twee haakse bochten, zodat de draad, over zekere afstand verschoven, in dezelfde richting doorloopt).

## 5. DATASTRUCTUREN

### 5.1 Data-representatie

Om te kunnen besluiten hoe de data in de datastructuur gerepresenteerd moeten worden, dient er eerst een overzicht gemaakt te worden van wat er nodig is.

Er zijn twee typen layoutelementen.

- Draadstukken. Daarvan zijn de volgende gegevens nodig:
  - De coördinaten. Om de posities van alle zijden te weten zijn er vier coördinaten (of twee paar) nodig.
  - Wanneer vóór de compactie colouring is toegepast, mag er niet meer van worden uitgegaan dat de horizontale draden zijn uitgevoerd in interconnectie: de laag dient expliciet bekend te zijn. Ze hoeft niet bij elk draadstuk vermeld te worden; het is genoeg wanneer dat gebeurt waar de draad begint of eindigt op een gridpunt.
  - Ook voor het netnummer van de draad is het voldoende als het bij het gridpunt te vinden is.
- Via's. Hiervan willen we kunnen beschikken over:
  - De coördinaten naar een referentiepunt (het centrum) van de via. Omdat via's vaste afmetingen hebben is één coördinatenpaar afdoende.

Om een contour adequaat te kunnen weergeven is er slechts een lijst nodig van paren x-coördinaten met bij ieder paar een geldige y-coördinaat.

De eenvoud van de layoutelementen maakt reductie van de gegevens per element onmogelijk: met minder kan het niet. De data worden eenvoudig als integers gerepresenteerd.

### 5.2 De datastructuur voor de bedrading

De compactie, zoals die hier is gerealiseerd, verandert de topologie van de bedrading niet wezenlijk. We maken hiervan gebruik door de compactie slechts toe te passen op de horizontale draden; de verticale worden op grond van de topologische informatie aan de horizontale aangepast. De datastructuur moet deze benadering mogelijk maken.

De eisen die we aan de datastructuur stellen zijn:

- Een layoutelement moet snel te vinden zijn aan de hand van een tracknummer en een x-coördinaat.
- Details\* moeten goed weergegeven kunnen worden.
- De datastructuur moet onafhankelijk zijn van de design rules.
- De topologische informatie moet in de datastructuur vervat zijn.

Er is een tweedimensionaal array gekozen waarvan de elementen corresponderen met de originele gridpunten van de router. De indices heten daarom "track" en "column". In een gridpunt kan een via liggen en er kunnen maximaal vier draden beginnen (of eindigen). Ieder array-element - ook wel "node" genoemd - kan dan ook vijf pointers bevatten: één naar de via en één naar elke draad. Ontbreekt in dit gridpunt de via of een of meer draden, dan zijn de overeenkomstige pointers van de node nil.

---

\* De ondergrens van de detaillering ligt in de representatie met behulp van integers. Meer details zijn mogelijk door met vergrote layoutelementen te manipuleren, die voor gebruik verkleind worden.

In het via-record staan de coördinaten van het centrum van de via en - enigszins overbodig - het netnummer. Ieder draad-record, "branch" geheten, heeft plaats voor de laag van de draad, zijn netnummer en een pointer die wijst naar het begin van de dubbel gelinkte list waarin de draadstukken worden opgeborgen.

De draadstukken worden gesorteerd in de list opgenomen. Wanneer we bijvoorbeeld de list van de rechter branch van een node ingaan, dan vormen de achter elkaar gevonden draadstukken één aaneengesloten pad dat uitkomt bij de linker branch van de rechter buur node van het uitgangspunt.

De records waarin de draadstukken worden opgeslagen, bevatten naast een "next" en een "previous" veld de coördinaten van de linker-beneden en van de rechter-boven hoek van het draadstuk.

Voldoet de datastructuur nu ook aan de gestelde eisen?

Er is geen compactie in de x-richting. De x-coördinaat van de columns verandert dus niet. Wordt er een layoutelement gezocht aan de hand van track en x-coördinaat dan is het nummer van de dichtsbijzijnde column onmiddellijk uit de x-coördinaat te bepalen. Track en column nummer zijn zo bekend en er kan meteen in een "node" van het array de "branch" list afgelopen worden, tot het draadstuk met de juiste coördinaat is gevonden. Aangezien zo'n branch slechts de afstand tussen twee columns overspant, is het aantal elementen in de list klein en het gezochte element snel gevonden.

De datastructuur begrenst op geen enkele wijze het aantal draadstukken dat tussen twee naburige "nodes" wordt neergelegd: de draad kan eindeloos gedetailleerd worden door elementen aan de "branch"-list toe te voegen.

De topologie is duidelijk in deze datastructuur gerepresenteerd en de opzet ervan heeft geen relatie met welke design rule ook.

We mogen dan ook concluderen dat aan de gestelde eisen wordt voldaan.

### 5.3 De datastructuur voor de contouren

Voor de contouren kunnen de volgende eisen geformuleerd worden:

- Voor iedere bedradingslaag moet een contour beschikbaar zijn.
- Elke contour moet een ononderbroken grens vormen.
- Op de vraag welke y-coördinaat de grens voor een zekere bedradingslaag bij een gegeven x-coördinaat heeft moet één en slechts één antwoord gevonden worden.
- Ten behoeve van de stretching moeten alle contouren die bij de compactie berekend worden bewaard blijven.

Noemen we een set contouren, voor iedere bedradingslaag één, een "outline". Voor elk track is er één outline.

Om alle outlines op te bergen is een dubbel linked list gebruikt, "contourlist" geheten. Elk record daarvan bevat - naast de gebruikelijke pointers "previous" en "next" - een tracknummer en de outline die bij dit track hoort. Omdat de algoritmen track voor track werken zal er niet vaak in de linked list gezocht hoeven te worden. In een linked list zijn niet meer data aanwezig dan nodig zijn en dat weegt in dit geval op tegen het snelle zoeken dat een array mogelijk maakt.

Zoals gesteld is in elk record van de contourlist een outline aanwezig. Alle contouren van één outline - bij deze implementatie twee - zijn geïntegreerd in één datastructuur, die bestaat uit een eendimensionaal array. De index heet "column", een element "contournode". Iedere contournode heeft een pointer naar één via en voor iedere bedradingslaag naar een linker en een rechter tak; vijf pointers in totaal. Anders dan bij de layout datastructuur zijn deze pointers nooit nil, maar wijzen ze steeds naar valide data. Dat is een van de maatregelen om ononderbroken contouren te realiseren.

De records waar de vijf pointers van een contournode naar toe wijzen zijn van dezelfde types als bij de datastructuur voor de layout. Evenals bij deze laatste is het ook hier zo dat wanneer men in een contournode een rechters tak van zekere laag ingaat, men in de rechts ervan gelegen contournode in de linkertak van dezelfde laag eindigt.

Vergelijkt men de datastructuur met de geformuleerde eisen, dan blijken alleen de eerste eis - voor iedere bedradingslaag moet een contour beschikbaar zijn - en de laatste - ten behoeve van de stretching moeten alle contouren die bij de compactie berekend worden bewaard blijven - gerealiseerd te zijn. Voor de andere eisen - dat elke contour een ononderbroken grens moet vormen en dat elke contour een éénwaardige functie moet zijn, gedefiniëerd voor alle  $x$  over de lengte van het hele kanaal - is een goede datastructuur kennelijk niet voldoende: ook de algoritmen moeten de vervulling van deze eisen zeker stellen.

## 6. COMPACTIE MET BEHULP VAN CONTOUREN

De topologie van de bedrading wordt door de compactie intact gelaten. Dat maakt het mogelijk de compactie slechts toe te passen op de horizontale draden; de verticale worden op grond van de topologische informatie aan de horizontale aangepast.

Het compactie-algoritme:

1. De eerste outline wordt geïnitieerd. Daartoe wordt één van de horizontale kanaalranden in de outline gekopieerd. Dat gebeurt dus één maal voor iedere contour in de outline. De outline die op een zeker moment van belang is, heet "de actuele outline", ter onderscheiding van alle andere. Zoals alle outlines bevat de actuele outline voor iedere laag een contour.
2. Het eerste track, geteld vanaf de kanaalrand die voor de initialisatie is gebruikt, wordt gekozen; de gecreëerde outline wordt de actuele outline.
3. Het track wordt afgelopen en elke draad die er voorkomt, wordt aangeschoven tot op isolatie-afstand van die contour uit de actuele outline die overeenkomt met de laag van de draad.
4. Het track wordt opnieuw afgelopen en alle via's worden aangeschoven. Zij moeten op isolatie-afstand blijven van beide contouren van de actuele outline; ze zijn immers in beide geleiderlagen aanwezig.
5. Indien dit een verticaal kanaal is en er is in dit track een aansluiting naar links of rechts het kanaal uit, dan wordt die aansluiting op het grid van het horizontale kanaal gelegd.
6. Het is mogelijk dat draden en via's door het aanschuiven contact verloren hebben. In de nodes van dit track worden de correcte aansluitingen hersteld.
7. In de contourlist wordt een nieuw element gegenereerd. De nieuwe outline wordt samengesteld uit de gegevens van de actuele plus de data van de aangeschoven draden. De nieuwe outline wordt de actuele outline.
8. Het volgende track wordt gekozen.
9. Punt 3 tot en met 8 worden herhaald totdat alle tracks behandeld zijn.
10. Uit de laatst gegenereerde outline wordt de nieuwe hoogte van het kanaal bepaald.

De bedrading waarvan wordt uitgegaan bestaat uit eenvoudige, rechte stukken draad. Men kan het verloop van zo'n draad opvatten als een wiskundige functie.\*

$$\forall (x \geq x_l \cap x \leq x_r): y = C$$

Het is meteen duidelijk dat het om een eenwaardige functie gaat: voor iedere  $x$  waarvoor de functie gedefiniëerd is bestaat er één valide waarde van  $y$ . Deze eigenschap blijft na compactie behouden. Compactie kent wel nieuwe waarden toe aan de  $y$ -coördinaat, maar daarbij gaat het om vervanging: het blijft één waarde van  $y$  voor elke  $x$ . Dat is van belang voor de stretching.

De orde van de compactie is laag. De bewerkingen die noodzakelijk zijn om één draadstuk aan te schuiven zijn uitsluitend afhankelijk van de direct nevenliggende draad. Uit de beschrijving blijkt dat het algoritme volledig onafhankelijk is van de positie in het track, of het aantal tracks. Kortom: voor ieder draadstuk wordt een vast deel van de

---

\* Welke functies er toegestaan zijn wordt bepaald door het layoutstelsel. Met name de eis dat ieder layoutelement uit rechthoeken moet zijn opgebouwd beperkt het aantal mogelijke functies sterk.

omgeving onderzocht. Daaruit kan geconcludeerd worden dat het algoritme lineair is in het aantal draadstukken.

Na compactie vertonen de kanalen een gecomprimeerde, sterk meanderende bedrading. Zowel het uitgangspunt voor de compactie, "auto-pin", afmetingen 764 X 696  $\mu\text{m}$ , als het resultaat, met als titel "compacted", afmetingen 644 X 636  $\mu\text{m}$ , is te vinden in de appendix. "Compacted" neemt nog slechts 77% van het originele oppervlak in beslag.

De bezwaren zijn echter duidelijk:

- Toename van de draadlengte en daardoor van de weerstand wat vooral bij de polysilicium bedrading erg hinderlijk is.
- Toename van de capaciteit, eveneens het meest bezwaarlijk bij de polysilicium bedrading.
- Het grote aantal bochten beïnvloedt de levensduur van het IC nadelig.
- Het enorme aantal rechthoeken waaruit de layout is samengesteld vereist een even enorme data-opslag en is zeer tijdrovend bij de IC fabricage.

Deze bezwaren, die alle zijn te herleiden tot de overmaat aan bochten in de bedrading, dienen door toepassing van stretching te worden ondervangen.

## 7. STRETCHING

### 7.1 De grote lijnen van het algoritme

Bij het stretchen wordt iedere draad opnieuw in de layout neergelegd, ditmaal met zo min mogelijk bochten.

Het doel van de stretching is het zoveel mogelijk verwijderen van bochten uit de gecompecteerde layout, zonder dat daarvoor extra oppervlak nodig is.

De stretching werkt, net als de compactie, track voor track, maar in de omgekeerde volgorde. De rangnummers die aan de tracks gegeven worden zijn dezelfde als bij de compactie: wordt er aan het eerste track gerefereerd dan is het track bedoeld dat het eerst gecompecteerd werd. De stappen van het algoritme zijn:

1. Er wordt één additionele outline geïnitieerd. Daartoe wordt de nieuwe kanaalrand waar de compactie bij eindigde in de outline gekopieerd. Dat gebeurt dus één maal voor iedere contour in de nieuwe outline. Van de outlines die als resultaat van de compactie ter beschikking staan, wordt die gekozen die het voorlaatste track beschrijft.
2. Het laatste track wordt gekozen.
3. Het track wordt afgelopen en elke draad die er voorkomt, wordt gestretched in de ruimte die door de beide outlines wordt omsloten.
4. Het track wordt opnieuw afgelopen en alle via's worden naar de gestretchte draden geschoven.
5. Indien dit een verticaal kanaal is en er is in dit track een aansluiting naar links of rechts het kanaal uit, dan wordt de gestretchte draad correct met die aansluiting verbonden.
6. Het is mogelijk dat draden en via's door het verschuiven contact verloren hebben. In de nodes van dit track worden de correcte aansluitingen hersteld.
7. In de contourlist wordt de voorgaande outline gezocht. De additionele outline wordt aangevuld met de data van de gestretchte draden.
8. Het voorgaande track wordt gekozen.
9. Punt 3 tot en met 8 worden herhaald totdat alle tracks behandeld zijn.

In het volgende wordt de nadere uitleg gegeven die punt 3 behoeft.

### 7.2 De basisprincipes van stretching

De vraag rijst of het altijd mogelijk is het resultaat van de compactie te stretchen. Het antwoord - ja - wordt in de volgende paragrafen toegelicht. Daarna volgt een bespreking van de stretching in detail.

Het beginsel waar stretching gebruik van maakt laat zich als volgt omschrijven: in een afgebakende ruimte waarbinnen een draad heeft gelopen onder in acht name van alle design rules kan, wanneer de ruimte niet verkleind wordt, altijd een nieuwe draad van dezelfde laag getrokken worden.

Kiezen we als uitgangspunt voor deze uiteenzetting het laatste gecompecteerde track van het gecompecteerde kanaal. De hoogte van het kanaal is al tot de minimale waarde teruggebracht. Het laatste track ligt ingeklemd tussen de nieuw berekende kanaalrand en de outline van het voorgaande track. De bedrading in het laatste track voldoet aan de design rules; daar heeft het compactie-algoritme voor gezorgd. Ook de nieuwe kanaalrand is berekend in overeenstemming met de design rules. We constateren dat de ruimte tussen de kanaalrand en de outline voldoende is om de bedrading van het laatste track plaats te bieden. Volgens het bovenstaande principe kunnen we de bedrading van

het laatste track wegnemen en opnieuw neerleggen, in precies dezelfde ruimte, ditmaal niet maximaal aangeschoven tegen de contour van het voorlaatste track, maar met een minimum aan bochten. In het ongunstigste geval is de ruimte zó beperkt dat de vorm van de draad volledig vastligt en dan vinden we de originele draad terug. In het algemene geval zijn er slechts enkele plaatsen waar de kanaalrand de bedrading tot op de isolatie-afstand benadert en bestaat er voor het overige een overschot aan ruimte. De keuzevrijheid die dat overschot biedt, wordt benut om de draad een minder bochtig verloop te geven.

De oppervlakte die het voorlaatste track ter beschikking staat, wordt begrensd door de tracks ter weerszijde ervan. De bedrading van het laatste track bevond zich vóór het herleggen op zo kort mogelijke afstand van het voorgaande track; door opnieuw leggen kan ze op dezelfde afstand of verder van het voorgaande track af komen te liggen. De ruimte voor het voorgaande track is dus vergroot, althans niet verkleind, door het stretchen van het laatste track, wat het stretchen van het voorlaatste track vergemakkelijkt.

### 7.3 De voorwaarden voor stretching

In de bedrading van een gecompacteerd kanaal worden, zoals uit voorgaande blijkt, één voor één de draden weggenomen en opnieuw gelegd. Bezien we dat iets meer in detail. Er is één draad verwijderd en nu moet in het daardoor ontstane pad een nieuwe draad geconstrueerd worden. De vraag doet zich voor: kan het stretching algoritme in ieder willekeurig pad een nieuwe draad genereren?

Daarvoor moet het pad aan twee voorwaarden voldoen:

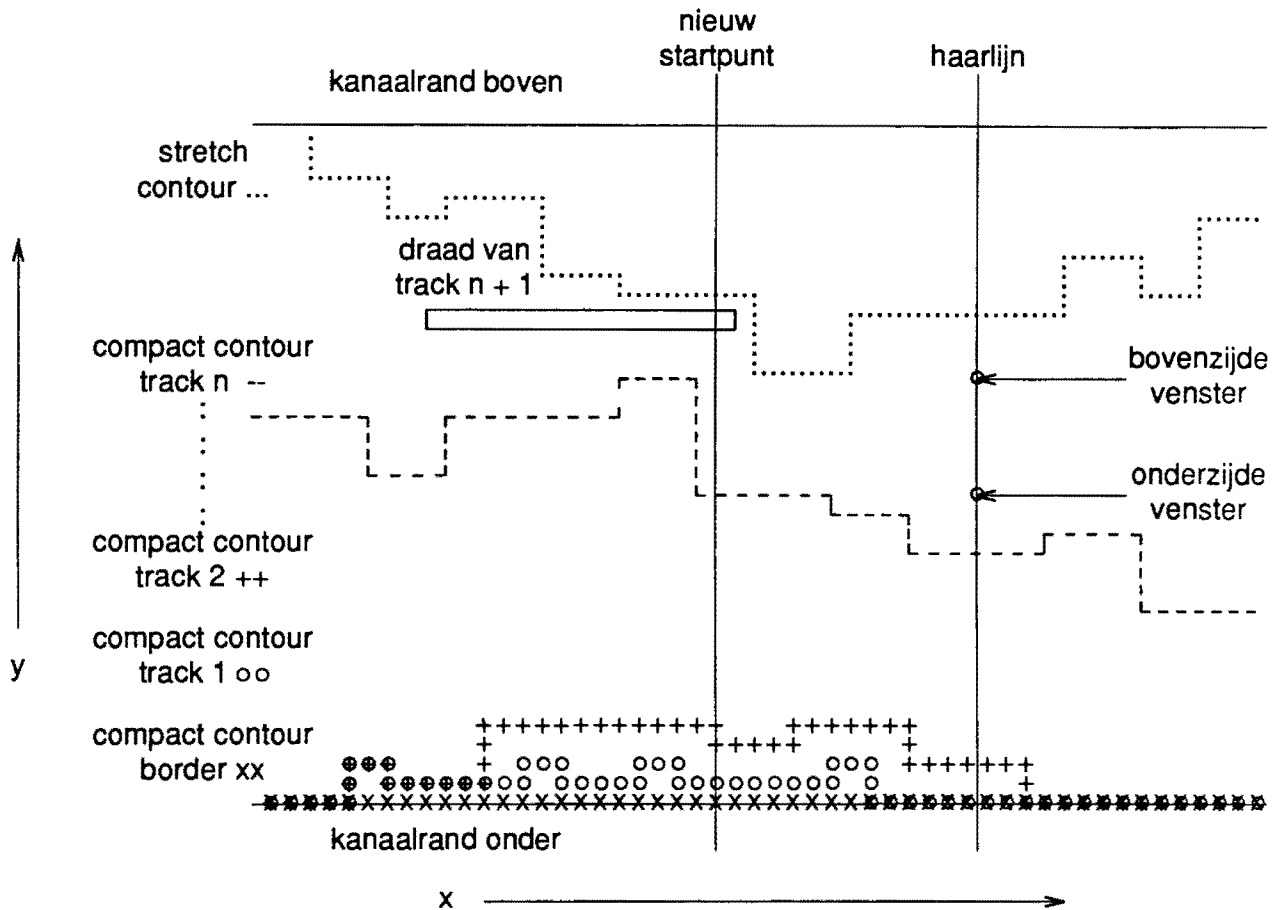
- Het pad moet zijn ontstaan door wegname van een draad die voldeed aan alle design rules. De nieuwe wordt gelegd, rekening houdend met de design rules. Dat houdt in dat de breedte van het pad op zijn minst eenmaal de minimale draadbreedte plus twee maal de minimale isolatie-afstand is. Zou, doordat de oorspronkelijke draad niet aan de design rules voldeed, de breedte van het pad ergens smaller zijn, dan faalt de stretching.
- De horizontale delen van de weggenomen draad mogen - afgezien van de aansluitingen met de naaste burens die één draadbreedte groot zijn - elkaar niet overlappen. De reden is dat de draad die bij het stretchen gegenereerd wordt, bestaat uit achter elkaar liggende horizontale delen, verbonden door verticale stukken. Een pad dat terug buigt kan hiermee niet bedraad worden. Ieder pad dat vrijkomt door het verwijderen van een draad die gerepresenteerd kan worden door een eenwaardige functie, kan door de stretching gevolgd worden. Zoals uit hoofdstuk 6, "compactie" blijkt, voldoen de compactie resultaten hier aan.

### 7.4 Stretching: de details

#### *Stretching per draaddeel.*

De bedrading wordt per track gestretched. Het track wordt van links naar rechts afgelopen en de column waarbij een draad begint wordt onthouden. De draad wordt gevolgd totdat hij ophoudt of van laag verandert. Een draad die gestretched wordt ligt daardoor over de hele lengte, die in één keer behandeld wordt, in dezelfde laag. Over het gevonden traject wordt de aanwezige draad door een nieuwe vervangen. Blijkt het noodzakelijk te zijn in de nieuwe draad een jog te maken, dan wordt het verticale deel van de jog in eerste instantie overgeslagen. De draad loopt dus door tot op isolatie-afstand van de hindernis die niet zonder bocht gepasseerd kan worden en begint daar opnieuw, op een andere hoogte in het kanaal. De aanvang van het nieuwe deel ligt precies één





Figuur 6. Stretching

draadbreedte meer naar links dan het punt waar het vorige deel ophoudt. Is de hele draad klaar, dan worden de verbindende verticale draadstukjes toegevoegd en de draad in de datastructuur opgeborgen. Daarna worden in het track een voor een de volgende draden gezocht en op dezelfde wijze gestretched, totdat het hele track behandeld is.

De draad ligt over de volle lengte die in één keer behandeld wordt in dezelfde laag. Daarom hoeft van elk van de outlines die het benutbare gebied omsluiten, alleen die contour beschouwd te worden die correspondeert met de laag waarin de draad ligt. Men dient zich te realiseren dat de draad weliswaar allerlei bochten maakt, maar hij ligt en blijft liggen op hetzelfde track: het track wordt in de y-richting gedeformeerd en maakt precies dezelfde bochten als de draad.

*De hulpmiddelen bij het stretchen (zie fig. 6).*

De methode maakt gebruik van een haarlijn ("scanline") die, loodrecht op het track staande, langs het hele traject van de draad wordt getrokken en daarbij voorbij de beide geselecteerde contouren glijdt. De minimum- en maximumwaarde van het snijpunt van de haarlijn met elk van de beide contouren wordt bijgehouden in de boven- en onderzijde (respectievelijk "WindowTop" en "WindowBottom") van een venster (Window). Het venster - dat is het verschil tussen de boven- en de onderzijde ervan - kan wél kleiner, maar niet groter worden. Zodra het te klein wordt, wordt een nieuw venster berekend.

### *Stretching met haarlijn en venster.*

Gaan we nu in op de stretching van een draad. De column waarbij de draad begint, wordt het startpunt. Daar worden de boven- en de onderzijde van het venster uit de respectievelijke contouren bepaald. De haarlijn glijdt naar rechts en telkens wanneer zij in een van de contouren een nieuwe y-coördinaat tegenkomt, wordt naar het venster gekeken. Uit de nieuwe y-coördinaat en de tegenoverliggende zijde van het venster wordt een tijdelijke venstergrootte berekend. Past de draad, inclusief de noodzakelijke isolatie, in het tijdelijke venster, dan gaat de haarlijn verder; daarbij wordt, wanneer het tijdelijke venster kleiner was dan het gewone venster, dat laatste verkleind alvorens verder te gaan.

Zo schuift de haarlijn langs het track. Twee gebeurtenissen kunnen haar stoppen:

- De draad, met isolatie, past niet in het tijdelijke venster. (Het gewone venster kan altijd een draad met isolatie bevatten.)

Vanaf het startpunt tot op isolatie-afstand van de te omzeilen hindernis wordt één draad neergelegd. De y-coördinaten van de draad blijven op isolatie-afstand van de boven- en onderzijde van het venster. Is het venster zó groot dat er in de y-richting nog vrijheid bestaat, dan wordt de draad zoveel mogelijk aan die kant gelegd waar de stretching in het kanaal begon.

Het nieuwe startpunt komt op een draadbreedte ter linkerzijde van het einde van de draad te liggen. De haarlijn wordt op het nieuwe startpunt gezet en het venster wordt opnieuw geïnitieerd. De haarlijn beweegt weer naar rechts.

- Het einde van de draad wordt bereikt. Vanaf het startpunt tot de laatste column van de draad wordt een nieuwe draad neergelegd op dezelfde manier als in het vorige punt.

## **7.5 Resultaten**

De resultaten zijn te zien in de appendix; blad 1 toont een kanaal in drie fasen (vóór compactie, na compactie en na stretching) en blad 5 bevat onder de titel "stretched" het stretchresultaat.

Vrijwel alle bochten zijn verdwenen; voorzover er nog resteren is het niet moeilijk te zien waarom ze nodig zijn.

De orde van de stretching is N, op grond van dezelfde argumenten als bij de compactie.

## 8. VERDERE OPTIMALISATIES

### 8.1 De aansluitingen van de verticale kanalen op de horizontale

#### 8.1.1 Inleiding

Er doet zich een probleem voor wanneer een verticaal kanaal verbindingen heeft met het boven- of benedenkanaal.

Binnen elk kanaal wordt gebruik gemaakt van een grid dat in dit hoofdstuk zal worden aangeduid met "het lokale grid". In de gehele layout vigeert een ander grid waaraan gerefereerd zal worden met "het globale grid". Het symbool  $G$  zal de horizontale grid-afstand van het globale grid aanduiden. Het grid van de horizontale kanalen komt overeen met dat van de gehele layout (het globale grid).

Verder is met "de bovenrand" van het kanaal steeds die rand van het kanaal bedoeld die tijdens de routing en compactie de bovenrand vormde, onafhankelijk van de wijze waarop het kanaal in de totale layout is geplaatst. Hetzelfde geldt, mutatis mutandis, ook voor de onderrand.

#### 8.1.2 Probleemstelling

De verticale kanalen worden één voor één gemaakt, te beginnen met het linker. Voordat ze in de layout geplaatst worden, roteert men ze  $90^\circ$  naar links (tegen de wijzers van de klok in). Beschouwen we het linker kanaal en nemen we aan dat er één of meer verbindingen met de horizontale kanalen (boven of beneden) bestaan. De hoogste waarde van de  $y$ -coördinaat binnen het kanaal correspondeert door de rotatie met de laagste  $x$ -coördinaat (0) van de gehele layout. Aansluitingen die op het globale grid liggen, hebben in de totale layout  $x$ -coördinaten die hele veelvouden van  $G$  van het punt ( $x = 0$ ) verschillen. Lokaal binnen het linkerkanaal gezien is dat dus op veelvouden van  $G$  ten opzichte van de maximale  $y$ -coördinaat van het kanaal.

Na compactie wordt de overtollige hoogte van het kanaal verwijderd. Begint de compactie van het linkerkanaal bij het laagste track, om bij het hoogste te eindigen, dan wordt na de compactie de maximale  $y$ -coördinaat - de referentie voor de externe aansluitingen van het kanaal - verkleint. Het veranderen van de referentie betekent dat de aansluitingen verschuiven ten opzichte van het globale grid.

In eerste instantie is dat probleem opgelost door als lokale referentie voor de externe aansluitingen de onderzijde van het kanaal ( $y = 0$ ) te nemen. Bij de plaatsing werd het kanaal zóveel vergroot dat de onderrand weer precies op het globale grid lag. De wetten van de statistiek gelden ook hier: per kanaal werd op deze wijze ongeveer de helft van  $G$  verspeeld; een verlies dat op 2 à 3% getaxeerd kan worden.

#### 8.1.3 De oplossing

Het doet in theorie niet ter zake of de tracks van beneden naar boven, dan wel van boven naar beneden behandeld worden, zolang de volgorde bij stretching maar tegengesteld is aan die bij compactie. Thans wordt bij de compactie uitgegaan van het track met het hoogste rangnummer. Aan het einde van de compactie wordt de positie van de kanaalrand waar de compactie bij eindigde opnieuw berekend. Nu wordt niet meer de  $y$ -coördinaat van de bovenrand verlaagd, maar die van de onderrand verhoogd. De referentie blijft daardoor onverlet; alle aansluitingen kloppen, ook zonder dat het hele kanaal verschoven wordt.

Voor de andere verticale kanalen geldt hetzelfde betoog. Er is slechts één verschil: in het algemeen zal de bovenrand van een verticaal kanaal na plaatsing in het geheel niet samenvallen met een gridpositie van het globale grid. Op het moment dat er met de

compactie van een kanaal begonnen wordt, zijn de resultaten van alle voorgaande kanalen bekend, evenals de breedte van de kolommen met cellen. Het is dus niet moeilijk om te berekenen met welke positie tussen het globale grid de bovenrand van het verticale kanaal na plaatsing correspondeert. Het verschil tussen de berekende positie en de eerstvolgende globale gridpositie wordt beschouwd als "offset". Door als lokale referentie niet de y-coördinaat van de bovenrand van het kanaal te nemen, maar daar eerst de offset af te trekken is er weer een correcte en niet aan verschuiving onderhevige referentie gevormd.

De verandering in de volgorde waarin de tracks worden gecompacteerd is op het allerlaatste moment gerealiseerd. Blad 6 van de appendix laat "nstretched" zien; het resultaat van deze aanpassing. Vergelijken met blad 5 - "stretched" - laat zien dat 3% oppervlakte reductie in dit geval haalbaar bleek.

De figuren in de appendix zijn, behalve blad 6, gebaseerd op de oorspronkelijke volgorde van bewerking; de resultaten in hoofdstuk 9 zijn met de nieuwe aanpak tot stand gekomen.

## 8.2 De gegevensoverdracht

Bij de evaluatie van het functioneren van het algoritme bleek een nog eenvoudiger aanpak zonder verlies van gunstige eigenschappen mogelijk. Vandaar de volgende aanbeveling.

De stretching heeft gegevens nodig over de draden die door de compactie zijn aangeschoven. Op dit moment staan die data op twee plaatsen ter beschikking: in de datastructuur waarin de draden zijn opgeborgen en in de contouren die van de compactie resulteren. Het is immers zo dat de draden tot op een vaste afstand van de contour geschoven worden en dat de loop van draden en contouren dus zeer nauw gerelateerd is. In de toekomst dient er dan ook een keus gemaakt te worden, vandaar dat de voor- en nadelen van beide informatiebronnen geïnventariseerd worden.

- Het gebruik van contouren heeft als voordeel dat de daarin aanwezige records elkaar wat betreft de x-coördinaat niet overlappen. Bij het gebruik van de informatie uit de draadrecords moet in het oog gehouden worden dat ter plekke van één x-coördinaat maximaal drie elkaar overlappende records mogen voorkomen (bij "jogs"). Op zoek naar de juiste y-coördinaat moeten er dus altijd meerdere records onderzocht worden, wat enigszins onhandig is.
- Het bewaren van de contouren die bij de compactie gemaakt worden gebeurt uitsluitend ten gerieve van de stretching: het is de stretching die per track een volledige set contouren behoeft.

Een niet onaanzienlijk deel van de tracks is leeg. Bij het maken van de contouren moeten ter plaatse van deze lege delen de data uit een vorige contour gekopieerd worden. Zelfs wanneer er wel een draad op het track ligt is het nodig de contouren van de andere geleidingslagen uit hun voorgangers te dupliceren. Dat betekent - naast een onnodig grote data-opslag - ook het overbodig hanteren van gegevens. Voor de compactie zelf kan worden volstaan met één set contouren, waarvan alleen die contour wordt aangevuld die correspondeert met de laag van de aangeschoven draad. Aanvulling hoeft dan alleen maar te gebeuren over de lengte van de aangeschoven draad, niet over de hele lengte van het track.

Dit overziende suggereer ik uit te gaan van de aangeschoven draden.

### 8.3 Het compactie-algoritme op het laagste niveau

Het compactie-algoritme op het laagste niveau is op zich een goed algoritme, alleen strompelt het wat ongelukkig van het ene gridpunt naar het volgende. Het zou een verbetering betekenen wanneer het op dezelfde wijze aangeropen zou worden als het stretching-algoritme: in één keer wordt een hele draad - voor zover die niet overgaat naar een andere laag - behandeld. Het zou een aantal initiaties schelen en tijdwinst opleveren.

## 9. RESULTATEN

Voor de resultaten zij allereerst verwezen naar de bijlage; deze tekeningen zijn overtuigender dan veel cijfers.

Er zijn een aantal tests gedaan met het programma. Daarbij zijn er twee varianten van het programma gebruikt.

Allereerst een versie waarbij de compactie buiten werking is gesteld, "NoComp". De uitschakeling is gebeurd door de aanroep van de procedure "squeeze" te verwijderen. Om ervoor te zorgen dat er dan nog steeds valide layouts gegenereerd worden, is daarbij uitgegaan van een versie waarbij de verticale kanalen zonedig verschoven worden om op correcte wijze contact te maken met de horizontale kanalen. Alle niet gecompacteerdde resultaten zijn met dit programma gemaakt.

De tweede versie is de normale uitvoering. De volgorde waarin de tracks gecompacteerd worden, is zoals besproken in paragraaf 8.1. Verschuiven van de verticale kanalen is hierbij overbodig.

De oppervlakten van de layouts zonder compactie staan in kolom 3 van tabel 1. Dezelfde voorbeelden ná compactie zijn te vinden in kolom 4. In kolom 5 staat de reductie in procenten, met de ongecompacteerde layout oppervlakte als referentie.

In beide situaties is de oppervlakte van alle kanalen in de layout opgeteld;\* ze is te vinden in de kolommen 6 en 7 van dezelfde tabel. Ook deze oppervlakten zijn vergeleken, in kolom 8, en daar is te zien dat de praktijk de theorie achter zich laat. Is er theoretisch maximaal een compactie van de bedrading haalbaar van 33%, de voorbeelden die lager in de tabel staan dan dc1.dec laten hogere reductiepercentages zien. Een nadere bestudering van de tabel leert dat de hoge percentages vooral gevonden worden bij kleine layouts. De oorzaak moet gezocht worden in het verschuiven van de verticale kanalen dat noodzakelijk is wanneer er niet gecompacteerd wordt. Bij kanalen die niet hoog zijn betekent een verschuiving over 9  $\mu\text{m}$  (de gemiddelde afstand waarover verschoven wordt) al snel een verlies van 10%, wat de resultaten natuurlijk vertekent.

Ook wanneer we dit effect in gedachten houden mogen de resultaten goed genoemd worden.

Deutsch [7] noemt verbeteringen van 15-20% haalbaar. Hij vermeldt dat het voorbeeld dat hij uitwerkt zonder compactie 21% groter is, wat een reductie van ruim 17% impliceert. De cijfers hebben betrekking op kanalen en zijn dus te vergelijken met de gegevens onder "bedraad oppervlak". Daarbij zijn de belangrijkste design rules die hij gebruikt (draadbreedte, isolatie-afstand en via-grootte) hetzelfde.

---

\* Het is niet zo dat de som van de oppervlakte van de cellen en van de kanalen de oppervlakte van de hele layout uitmaakt. Er zijn meer zaken die extra oppervlak vragen: de feedthroughs, de lege ruimten boven de kolommen van cellen die niet de maximale lengte hebben en de voedingslijnen. Verder worden kanalen niet langer gemaakt dan nodig is, dus tussen het einde van een verticaal kanaal en het boven kanaal kan eveneens een lege ruimte aanwezig zijn.

OPPERVLAKTEREDUCTIE							
voorbeeld naam	aantal cellen	gehele layout			bedraad oppervlak		
		oorspr. in mm <sup>2</sup>	compac. in mm <sup>2</sup>	reductie in %	oorspr. in mm <sup>2</sup>	compac. in mm <sup>2</sup>	reductie in %
clip.dec	82	4.63	3.63	21	2.95	2.17	26
in7.dec	89	3.95	3.04	23	2.70	1.88	30
sqn.dec	76	3.83	2.91	24	2.54	1.78	30
rd84.dec	77	3.40	2.62	23	2.26	1.61	28
dc2.dec	77	3.69	2.89	22	2.30	1.67	27
alu2.dec	84	3.36	2.55	24	2.24	1.56	31
rd73.dec	61	2.59	2.00	23	1.57	1.12	28
risc.dec	77	2.71	2.05	24	1.71	1.15	32
fan.log	121	2.85	2.16	24	1.69	1.13	33
fx4	57	2.38	1.83	23	1.54	1.06	31
5xp1.dec	66	2.36	1.80	24	1.51	1.06	30
vg2.dec	60	2.36	1.79	24	1.47	1.00	32
alu3.dec	49	1.88	1.42	24	1.15	0.78	32
dc1.dec	24	0.802	0.598	26	0.449	0.302	33
wim.dec	22	0.764	0.574	25	0.397	0.262	34
fx6	18	0.532	0.396	26	0.285	0.179	37
fx5	24	0.569	0.451	21	0.290	0.203	30
f2.dec	10	0.338	0.235	30	0.166	0.101	39
xor5.dec	12	0.289	0.216	25	0.111	0.070	37
con1.dec	9	0.255	0.186	27	0.109	0.066	39
myex.dec	7	0.147	0.111	25	0.062	0.039	37
fx3	8	0.203	0.146	28	0.084	0.054	39
fx2	8	0.169	0.129	23	0.069	0.044	36
fx1	7	0.112	0.094	16	0.037	0.026	28

TABEL 1. Oppervlaktereductie

REKENSNELHEID							
rekentijd versus bedradingsoppervlak					rekensnelheid		
voorbeeld naam	rekentijd $\Delta t$ in s.	bedraad oppervlak $O_b$ in mm <sup>2</sup>	draad-delen N aantal	draad-lengte L in mm	opper-vlak/s $S_o$ in mm <sup>2</sup> /s	draad-deel/s $S_N$ in stuks/s	draad-lengte/s $S_L$ in mm/s
clip.dec	8.86	2.95	7118	68.0	0.322	779	7.43
in7.dec	7.65	2.70	6114	63.2	0.352	799	8.26
sqn.dec	7.29	2.54	5511	56.6	0.348	756	7.76
rd84	7.01	2.26	5643	57.1	0.322	805	8.15
dc2.dec	6.78	2.31	5436	54.6	0.340	800	8.05
alu2.dec	6.15	2.24	5035	50.6	0.356	800	8.00
rd73	5.18	1.57	3544	36.6	0.303	684	7.06
risc.dec	4.78	1.71	3905	40.2	0.357	817	8.41
fan.log	4.72	1.69	3234	35.1	0.358	685	7.45
fx4	4.37	1.53	3759	39.5	0.351	860	9.04
5xp1.dec	4.20	1.51	3241	33.6	0.367	785	8.14
vg2.dec	3.96	1.47	3028	30.1	0.372	765	7.79
alu3.dec	3.19	1.15	2517	25.8	0.345	756	7.76
dc1.dec	1.37	0.449	895	8.55	0.328	653	6.25
wim.dec	1.03	0.397	696	7.2	0.386	676	6.95
fx6	0.81	0.285	496	4.9	0.352	612	6.08
fx5	0.73	0.290	509	5.7	0.398	697	7.78
f2.dec	0.48	0.166	336	2.99	0.346	700	6.23
xor5.dec	0.42	0.111	196	2.2	0.264	467	5.31
con1.dec	0.39	0.109	164	1.98	0.279	421	5.01
myex	0.31	0.06	119	1.2	0.200	384	3.89
fx3	0.26	0.084	132	1.3	0.325	508	5.13
fx2	0.18	0.069	102	1.1	0.385	567	6.19
fx1	0.09	0.037	32	0.4	0.410	356	4.81
gemiddelde spreiding					0.34	672	6.97
spreiding (genormeerd)					0.05	148	1.36
					0.15	0.22	0.19

**TABEL 2.** Rekensnelheid

Om de claim van een lineair algoritme te bevestigen zijn van alle voorbeelden de relevante data in tabel 2 verzameld.

Het verschil in rekentijd tussen de compacterende en de niet compacterende variant, waaraan verder gerefereerd wordt als "de rekenduur" of "de rekentijd" en aangeduid met  $\Delta t$ , is de basis van tabel 2. Het is de tijd die daadwerkelijk aan compactie besteed is.

In de volgende kolommen is het oppervlak van de bedrading vóór compactie, het aantal draadstukken waaruit de gecompacteerd maar nog niet gestrekte bedrading op de



tracks bestaat en lengte van de horizontale bedrading vóór compactie te vinden. Uit deze gegevens en de rekenduur zijn rekensnelheden berekend. In kolom 6 staat hoeveel vierkante mm bedrading er per seconde verwerkt wordt ( $S_o$ ), in kolom 7 hoeveel draaddelen (uitsluitend draaddelen op de tracks) dat zijn ( $S_N$ ) en in kolom 8 de lengte draad die per seconde passeert ( $S_L$ ).

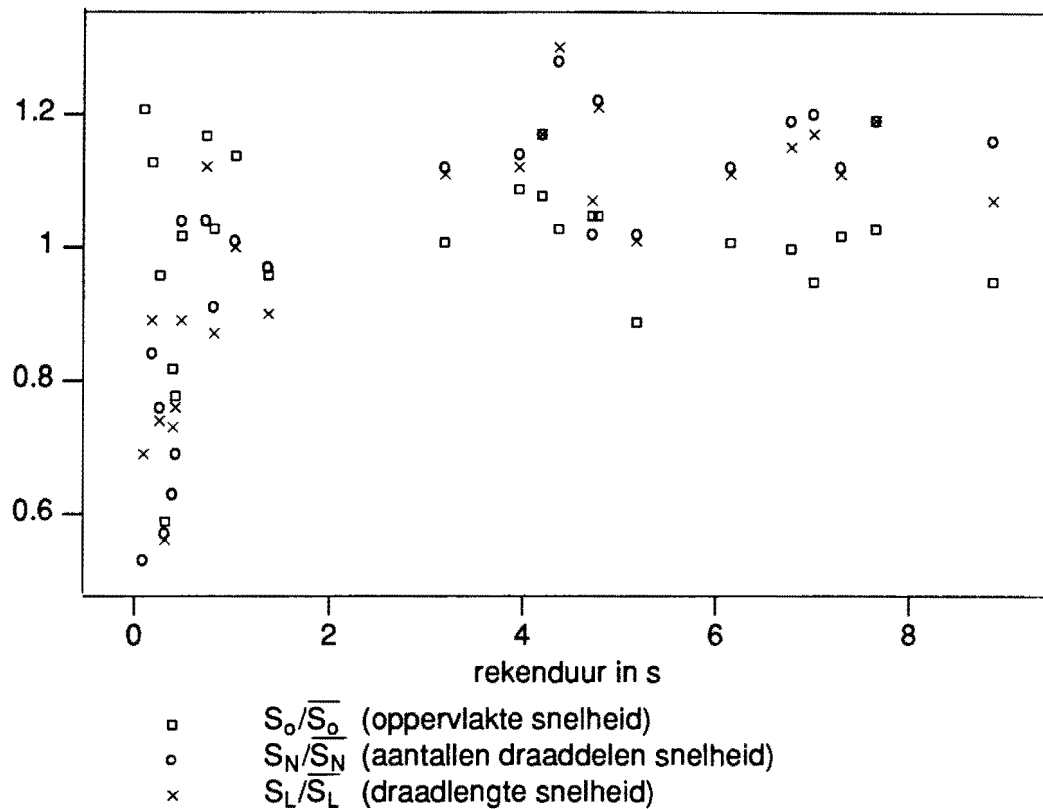
De tijd speelt in tabel 2 een belangrijke rol; daarom is de reproduceerbaarheid van de tijdmeting onderzocht. Twee voorbeelden, sqn.dec (groot) en xor5.dec (klein) zijn elk 10 maal gerout. De resultaten staan in tabel 3. De onzekerheid van de tijdmeting mag gering genoemd worden en is ternauwernood merkbaar in de gegevens van tabel 2.

REPRODUCEERBAARHEID VAN DE TIJDMETING						
(steekproefgrootte n = 10)						
voorbeeld	sqn.dec			xor5.dec		
oppervlak	4.63 mm <sup>2</sup>			0.289 mm <sup>2</sup>		
	met comp.	zonder comp.	verschil	met comp.	zonder comp.	verschil
maximum	13.22	6.16	7.50	0.85	0.51	0.44
minimum	12.72	5.72	6.56	0.75	0.41	0.30
gemiddelde	12.93	5.92	7.01	0.80	0.45	0.35
spreiding	0.170	0.149	0.23	0.028	0.029	0.040
spreiding (genormeerd)	0.013	0.025	0.028	0.035	0.064	0.073
alle tijden in seconden						

**TABEL 3.** Reproduceerbaarheid van de tijdmeting

Keren we na dit intermezzo terug naar de gegevens van tabel 2.

Bij een lineair algoritme zijn deze snelheden onafhankelijk van de grootte van het voorbeeld en dus van de rekenduur. Onder in de tabel zijn het gemiddelde, de spreiding, en de genormeerde spreiding van elke snelheid opgenomen. Daarbij kan opgemerkt worden dat, wanneer de kleinste voorbeelden - vanaf dc1.dec - buiten beschouwing worden gelaten, de genormeerde spreidingen verder afnemen tot 0.06. Bij deze kleine layouts met weinig te compacteren elementen wordt de statistische ruis storend.



**Figuur 7.** Genormeerde rekensnelheden tegen rekenduur

In figuur 7 zijn de genormeerde rekensnelheden uitgezet tegen de rekentijd. Voor ieder van de drie snelheden is zijn gemiddelde als norm genomen. De figuur illustreert duidelijk de toename van de ruis voor kleine rekentijden en suggereert constante snelheden bij grotere rekentijden.

Het kleine aantal testgevallen en vooral de beperkte omvang ervan maken enige reserve noodzakelijk bij het doen van uitspraken over de orde van het algoritme. De gepresenteerde data maken een lineair algoritme aannemelijk. Uit tabel 2 blijkt niet welke van de gebruikte grootheden: bedraad oppervlak, aantal draaddelen, of draadlengte, de nauwkeurigste voorspeller van de rekentijd is. Deze drie hangen onderling zo sterk samen dat dat ook van weinig belang is.

## 10. CONCLUSIES EN AANBEVELINGEN

### 10.1 Conclusies en aanbevelingen met betrekking tot het automatisch plaatsen van de pinnen aan de omtrek van de layout

De toegevoegde software ten behoeve van de automatische plaatsing van de pinnen langs de omtrek van de layout functioneert goed.

- Het gedrag van de router ten opzichte van de gebruiker is sterk verbeterd. Een tikfout in de invoer wordt niet langer afgestraft met het afbreken van het programma. Ook is het een groot gemak om niet meer bij elke run van het programma de hele lijst met gegevens van de in- en uitgangen foutloos te moeten intypen.
- De grote oppervlaktewinst die nu gemaakt wordt zal ten dele verloren gaan wanneer de gegevensuitwisseling met de architectuur-synthese en de floorplanner is geregeld. De winst die voor de deellayout verloren gaat komt dan de totale layout ten goede. Zolang echter de coördinaten van de pinnen langs de rand niet volledig worden vastgelegd zal er oppervlaktewinst geboekt worden op beide niveau's.
- Wanneer in de toekomst de gegevensuitwisseling tot stand komt, hoeft alleen het interactieve deel door een parser te worden vervangen. Alle overige vereiste functies zijn gereed.
- Het is dringend noodzakelijk dat er voor de gegevens die architectuur-synthese en floorplanner produceren - desnoods tijdelijke - afspraken worden gemaakt omtrent de plaats waar en het format waarin ze kunnen worden aangetroffen.
- Wanneer de coördinaten van de pinnen automatisch gegenereerd worden of door het opgeven van een volgorde van de pinnen tot stand komen, is het vermoedelijk gunstig de pinnen niet slechts in het midden van de layoutzijde te plaatsen, maar ze zo gelijkmatig mogelijk te spreiden langs de hele zijde: de verdeling van de bedrading over de kanalen zal erdoor verbeteren.

### 10.2 Conclusies en aanbevelingen met betrekking tot compactie en stretching

Zoals uit hoofdstuk 9 en uit de figuren in de appendix blijkt functioneert de combinatie van compactie en stretching zeer goed. Bereikte voordelen zijn:

- Het gekozen algoritme is overzichtelijk door de vaste volgorde van de stappen.
- Het gekozen algoritme breidt de mogelijkheden van de router uit: kanaalranden zijn niet langer beperkt tot rechte lijnen.
- De orde van de compactie is laag (N).
- De bezwaren die de compactie met zich meebrengt, veroorzaakt door het grote aantal bochten in de bedrading, worden door de stretching zeer goed ondervangen.
- De orde van de stretching is eveneens zeer laag (N).
- De nieuwe manier om bedrading in de verticale kanalen te laten aansluiten op die in de horizontale, levert een extra winst op van circa 3%.
- De oppervlaktereductie van de gehele behandelde layout schommelt rond de 23%.
- De beschikbare gegevens bevestigen de veronderstelling dat het algoritme lineair is.

Enkele aanbevelingen zijn:

- Er dient gekozen te worden welke data de stretching benut. Met name uit het oogpunt van rekensnelheid verdient het gebruik van de gecompecteerde draden de voorkeur boven het werken met de contouren die van de compactie resulteren, zoals thans gebeurt.
- Het compactie-algoritme moet in detail verbeterd worden.
- De toepassing van colouring vóór compactie wordt sterk aanbevolen.

## LITERATUUR VERWIJZINGEN

- [1] J. Mavor, M.A. Jack, and P.B. Denyer, *Introduction to mos lsi design*, Addison-Wesley publishing company, Edinburgh, 1982.
- [2] C.K. Wong, "An optimal two-dimensional compaction scheme," *Proceedings of the international workshop on paralel computing and VLSI, Amalfi, Italy*, pp. 205-220, 1984.
- [3] T. Hedges, W. Dawson, and Y.E. Cho, "Bitmap graph build algorithm for compaction," *Proc. third IEEE Int. Conf. on Computer-Aided Design (ICCAD'85), Santa Clara, California.*, pp. 340-342, 1985.
- [4] Y.E. Cho, "A subjective review of compaction," *Proceedings of the 22th Design Automation Conference*, pp. 396-404, june 1985.
- [5] A.E. Dunlop, "Slim-the translation of symbolic layouts into mask data," *Proceedings of the 17th Design Automation Conference*, pp. 595-602, june 1980.
- [6] N. Weste, "Virtual grid symbolic layout," *Proceedings of the 18th Design Automation Conference*, pp. 225-233, june 1981.
- [7] D.N. Deutsch, "Compacted channel routing," *Digest International Conference on Computer Aided Design*, pp. 223-225, 1985.
- [8] X.I.M. Timmermans, "Compac," *Thesis, University of Technology Eindhoven, Eindhoven*, 1983.
- [9] H.H. Tullemans, "Macro-cell generator for combinatorial logic," *Thesis, University of Technology Eindhoven, Eindhoven*, 1985.
- [10] D.N. Deutsch, "A "dogleg" channel router," *Proceedings of the 13th Design Automation Conference*, pp. 425-433, 1976.
- [11] P. Bruell and P. Sun, "a "greedy" three layer channel router," *Proc. third IEEE Int. Conf. on Computer-Aided Design (ICCAD'85), Santa Clara, California.*, pp. 298-299, 1985.
- [12] H.C. Du and K.C. Chang, "A new approach for layer assignment problem," *International Symposium on VLSI Technology, Systems and Applications; Proceedings of Technical Papers, Taipei, Taiwan.*, pp. 129-133, may 1985.
- [13] R.Y. Pinter, "Optimal layer assignment for interconnect," *Proceedings ISCAS 1982*, pp. 398-401, 1982.
- [14] L.P.P.P Ginneken van, "The predictor-adaptor paradigm," *Ph-d Thesis, University of Technology Eindhoven, Eindhoven*, 1989.

# APPENDIX

De appendix bevat zes bladen.

Blad 2 tot en met 6 betreffen hetzelfde voorbeeld; elke keer is er een algoritme veranderd of toegevoegd.

Blad 1 bevat driemaal hetzelfde kanaal, gelicht uit achtereenvolgens blad 3, 4 en 5.

Blad 1: Kanaal 3 in de drie opeenvolgende fasen in de compactie:

- orgineel (niet bewerkt)
- gecompacteerd
- gestretched

Blad 2: Het resultaat van het oorspronkelijke route programma, (log\_route) 1098 X 930  $\mu\text{m}$  groot.

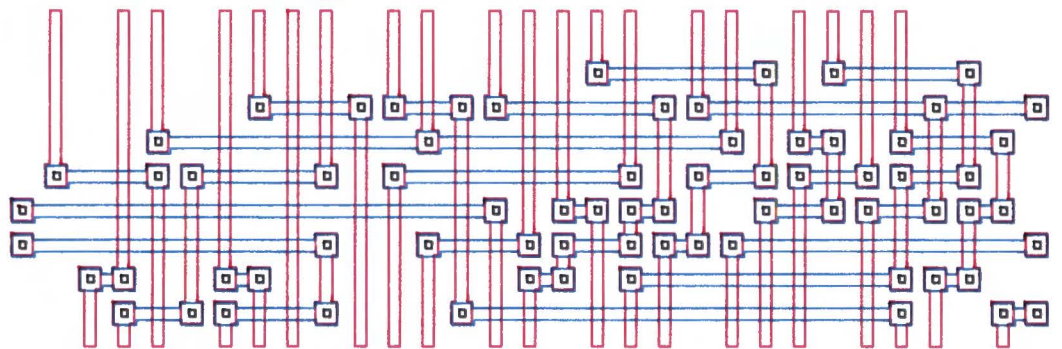
Blad 3: "auto-pin", 764 X 696  $\mu\text{m}$  groot. De pinnen aan de layout omtrek zijn automatisch geplaatst.

Blad 4: "compacted", 644 X 636  $\mu\text{m}$  groot. Hier heeft ook de compactie gewerkt.

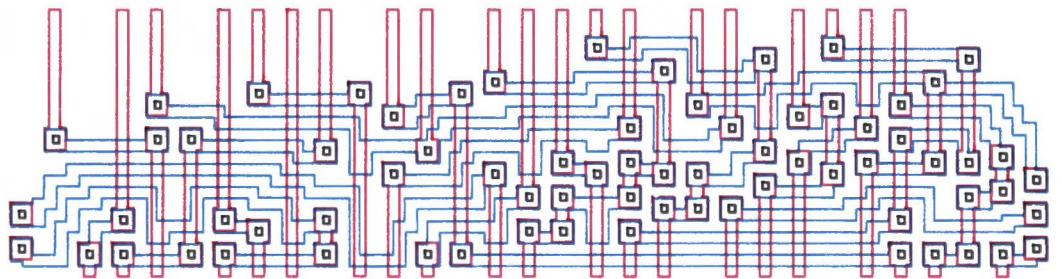
Blad 5: "stretched", 644 X 636  $\mu\text{m}$  groot. Alle uitbreidingen te zamen.

Blad 6: "nstretch", 622 X 636  $\mu\text{m}$  groot. Een "last minute" bijdrage. De volgorde van de tracks bij compactie en stretching is geïnverteerd. Daardoor is het niet langer nodig een heel verticaal kanaal te verschuiven om de aansluitingen naar het boven en beneden kanaal op het juiste grid te brengen, (zie 8.1) en verdwijnt de lege verticale strook links in sommige verticale kanalen.

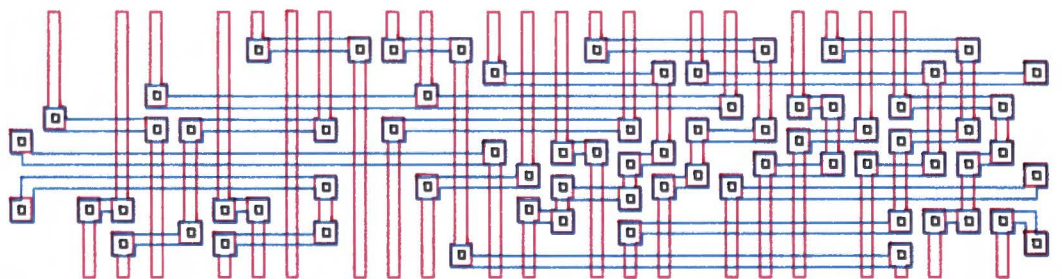
original



compact



stretched



Compound: demo 772um X 835um

Designer: huib\_h

Mon Jun 26 14:59:02 1989

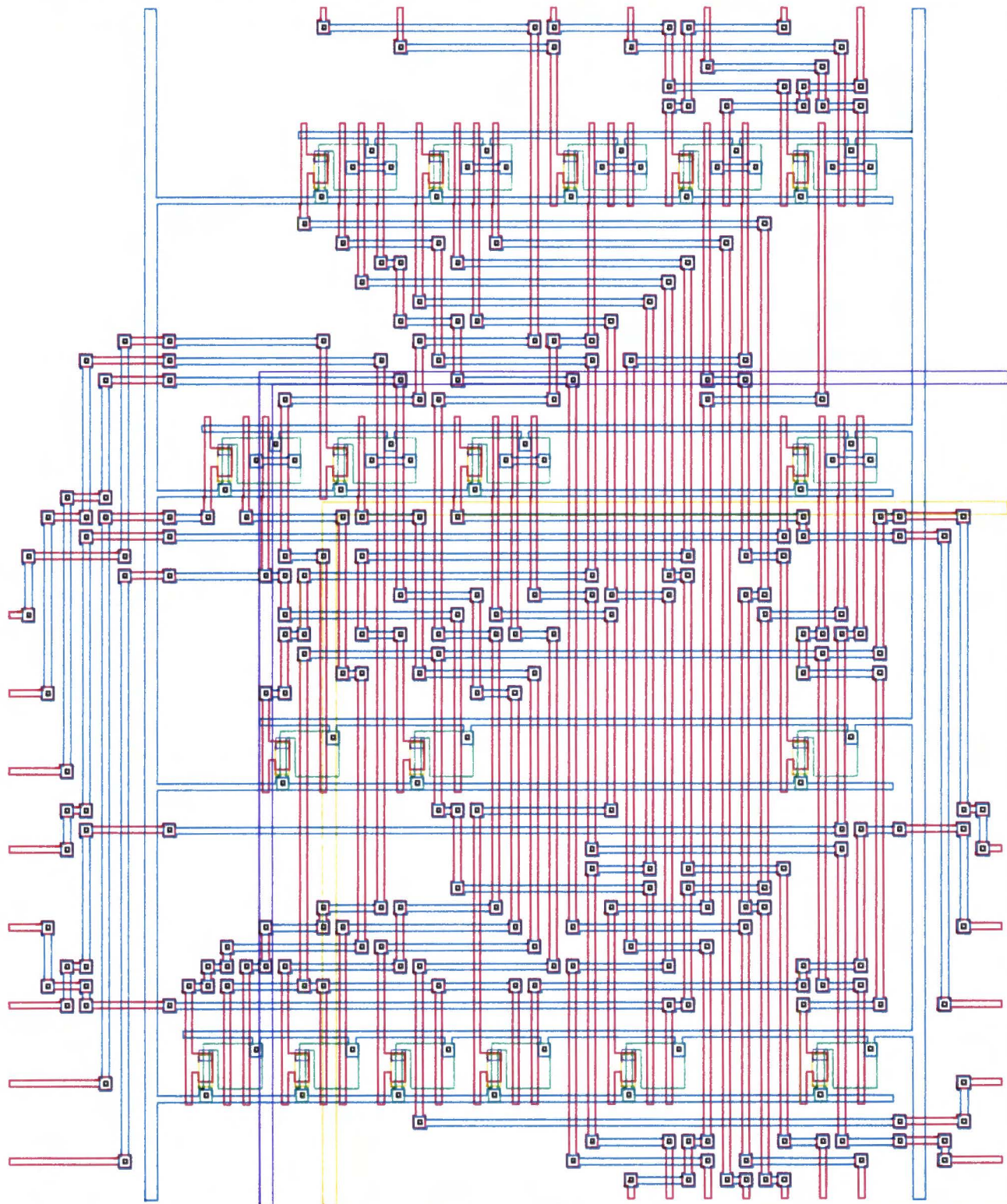
Window: 0, -3 772, 832

Scale: 1mm=3.99um

1um=0.25mm

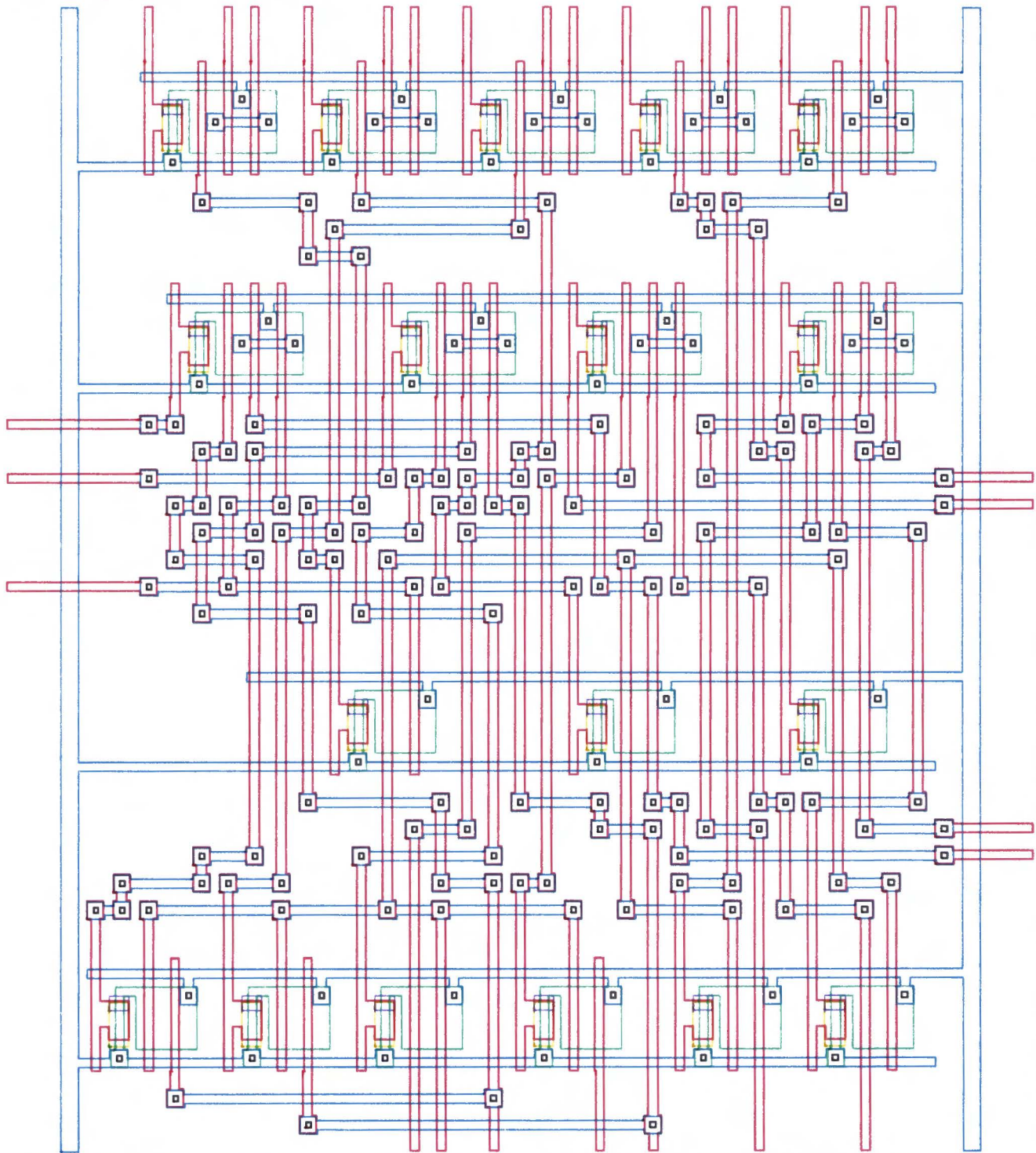
result of program "log\_route"  
(1098 x 930 um)

— dimensions of compound "autopin" (764 x 696 um)  
— dimensions of compound "stretch" (644 x 636 um)



Compound: henk 1098um X 930um Window: -9, -9 1300, 1050  
Designer: huib\_h Scale: 1mm=5.48um  
Mon Jun 26 13: 43: 22 1989 1um=0.18mm

auto-pin 764 x696

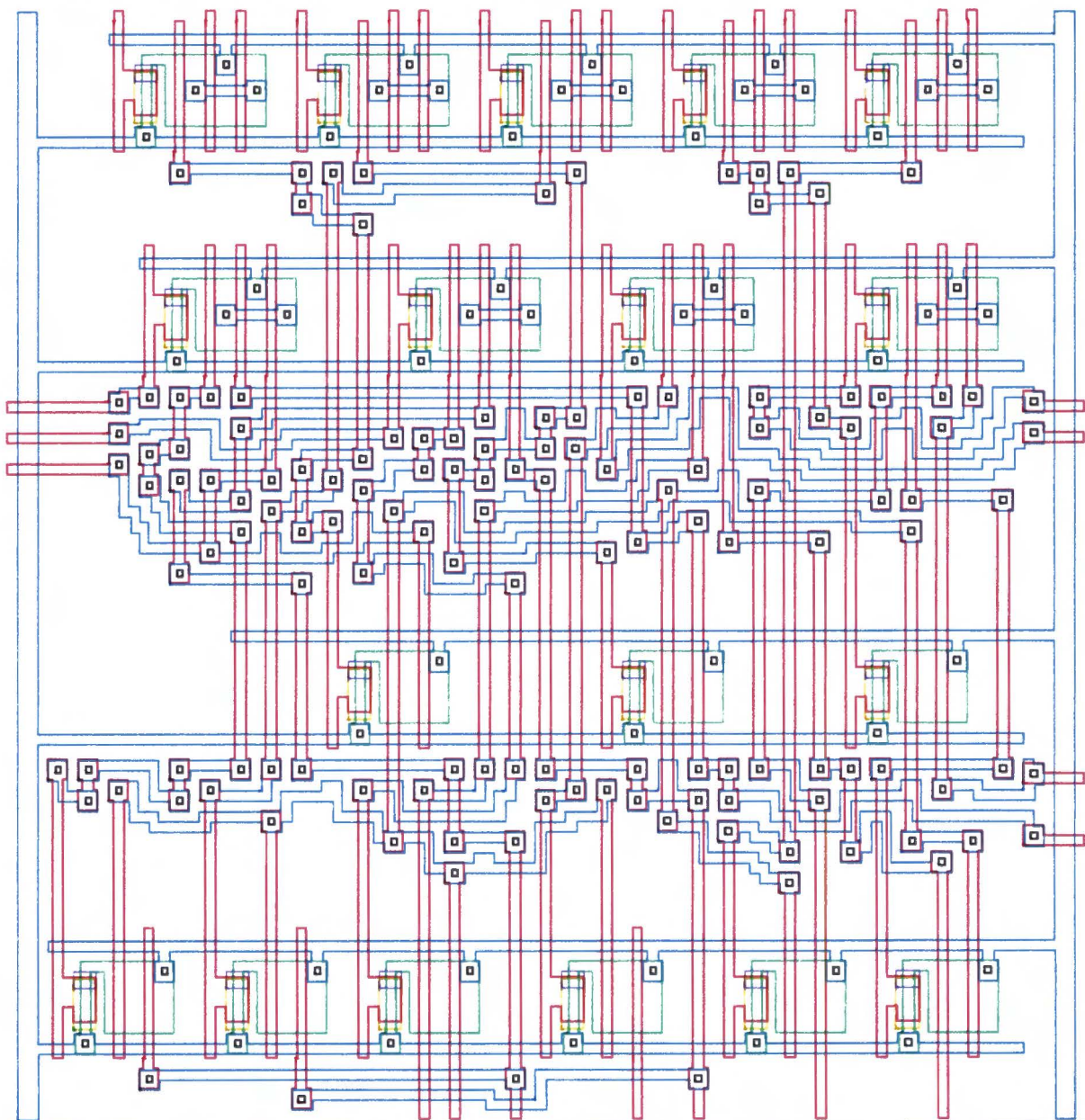


Compound: autopin 764um X 696um Window: -6, -5 826, 767  
Designer: huib\_h Scale: 1mm=3.99um  
Fri Jun 23 13: 12: 09 1989 1um=0.25mm



□ < original corner position of auto-pin

compacted 644 x 636 um



Compound: compactd 644um X 636um

Window: -3, -3 826, 767

Designer: huib\_h

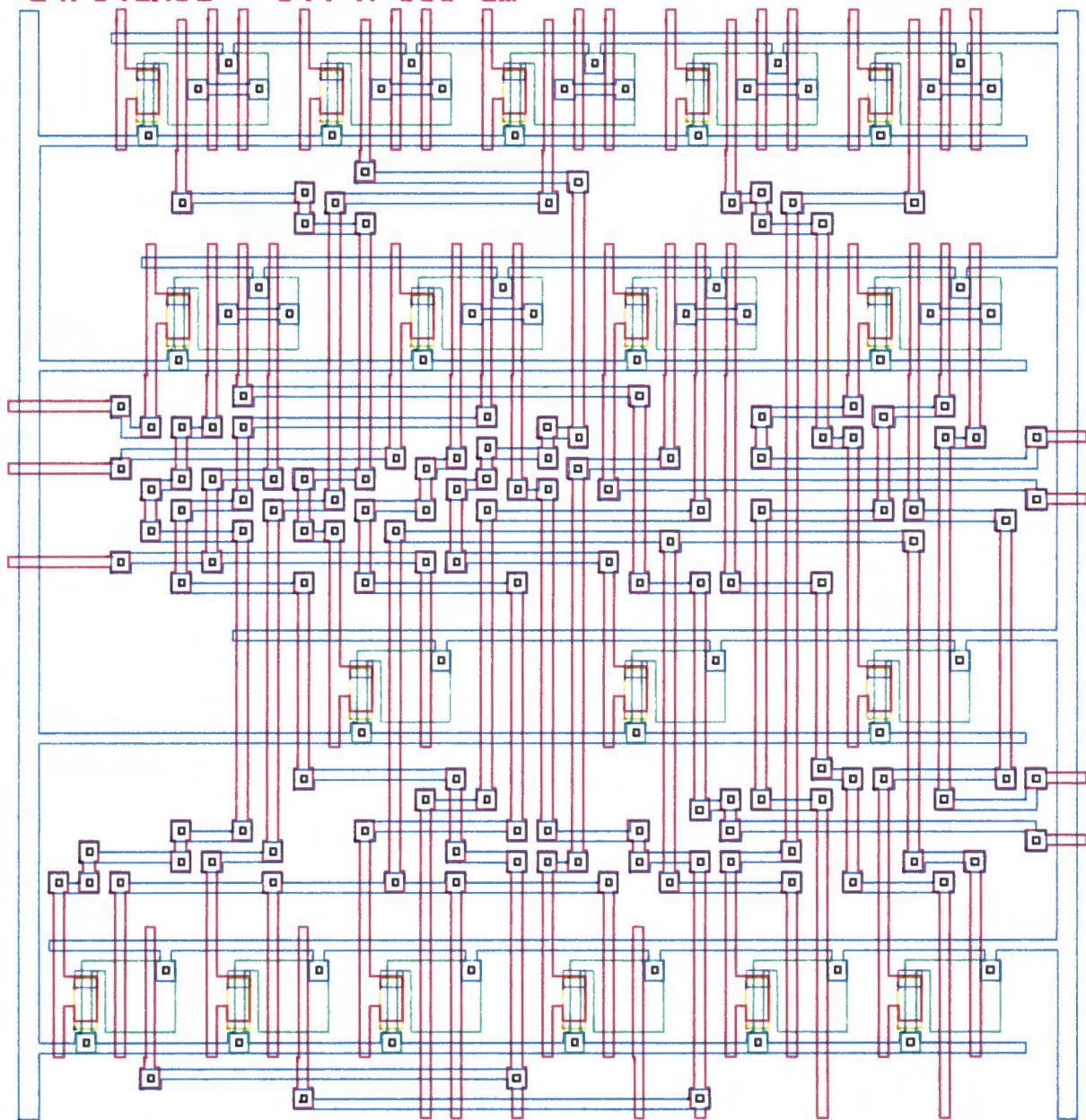
Scale: 1mm=3.98um

Fri Jun 23 13: 14: 41 1989

1um=0.25mm

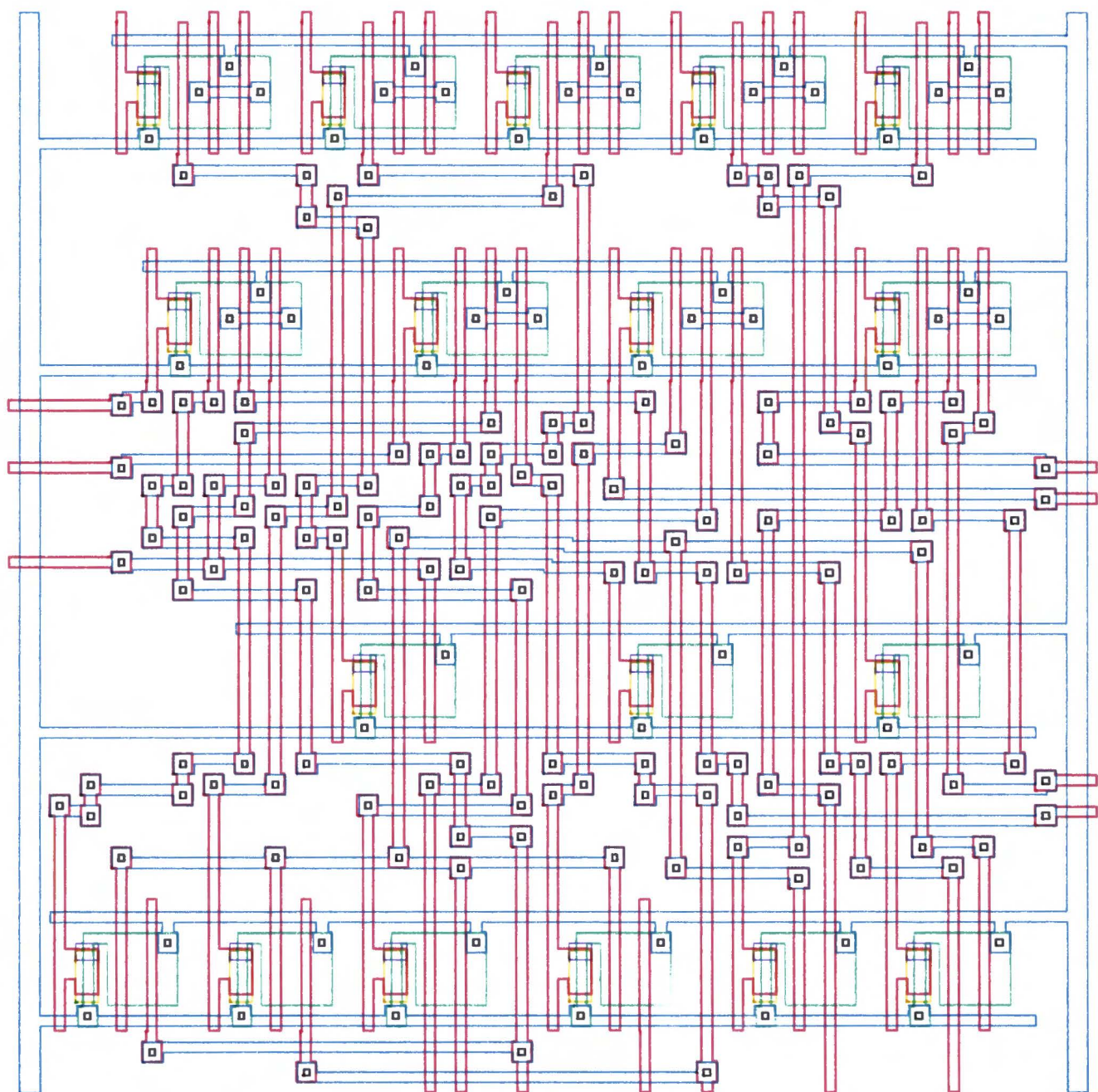
□ < original corner position of auto-pin

stretched 644 x 636 um



Compound: stretch 644um X 636um Window: -3, -3 826, 764  
Designer: huib\_h Scale: 1mm=3.97um  
Fri Jun 23 14: 05: 25 1989 1um=0.25mm

original corner position of autopin >  
compound "nstretch" (622 x 636 um)



Compound: nstretch      770um X 703um      Window: -3, -3 767, 700  
Designer: huib\_h      Scale: 1mm=3.64um  
Tue Jul 4 14: 19: 37 1989      1um=0.27mm