

# گزارش دوره کار آموزی



نام کارآموز: حمیدرضا بلوری کاشانی

کد دانشجویی: ۹۹۲۱۶۴۰۱۹

دانشکده: فنی و مهندسی

گروه آموزشی: مهندسی کامپیوتر

رشته: مهندسی کامپیوتر

مقطع: کارشناسی پیوسته

استاد راهنما: دکتر حقیقت دوست

محل کارآموزی: آزمایشگاه مدار و سیستم های مجتمع دانشکده فنی و مهندسی

دانشگاه شاهد

فصل: تابستان

سال: ۱۴۰۲

بہ نام خدا

## با تشکر از فراوان از

بنده بسیار خوشحال و خرسندم که زیر نظر استاد عزیز، جناب دکتر محمدباغر غزنوی قوشچی و اساتید دیگری چون آقای مهندس وهاب، خانم مهندس قائم نیا، آقای مهندس حسین نیا، آقای مهندس انصاری، دکتر پشتکوهی، خانم مهندس ایزدی نسب، آقای مهندس زیوری، و آقای مهندس وفایی نژاد بودم. دوره آموزشی بنده زیر نظر اساتید ذکر شده گذرانده شد که مباحث آموخته شده را در ادامه ذکر کرده ام.

تشکر دوباره و ویژه از خانم مهندس قائم نیا ( همگروه ) را دارم که علی رغم درگیری شغلی و آمادگی برای دفاع تز دکترای خود، با من جلسات رفع اشکال برگزار کردند.

همچنین از جناب دکتر غزنوی تشکر دوباره برای برگزاری جلسات متعدد رفع اشکال و راهنمایی و صبر و حوصله تشکر میکنم. ایشان علی رغم دارایی از فضائل علمی، از فضائل اخلاقی قابل توجهی برخوردار هستند که جای توجه و تقدیر دارد.

همچنین از آقای Tim Edwards ( سازنده نرم افزار طراحی دیجیتال به نام Magic ) کمال تشکر را دارم که به سوال بنده در خصوص نرم افزاری که نوشتند، به سرعت جواب دادند.

# ارزیابی

به نظرم بنده تجربه بسیار خوبی بود به چند دلیل:

1. با فرصت های دنیای الکترونیک آشنا شدم
2. با ترند های دنیای الکترونیک آشنا شدم
3. با نرم افزار های طراحی EDA در حوزه الکترونیک آشنا شدم
4. چند زبان برنامه نویسی یادگرفتم ( Python و TCL )
5. از دانسته های تئوری خود در تمرین ها و پروژه های محول شده استفاده کردم
6. فرصت رفع اشکال با استاد کار آموز وجود داشت
7. همکاری تیمی با افرادی که در حوزه الکترونیک بودند، داشتم

# چکیده

دوره کارآموزی سپری شده، فراز و نشیب های زیادی داشت،

ابتدای دوره به آموزش و آشنا کردن افراد با تکنولوژی ها و مفاهیم اولیه سپری شد. دوره آموزشی اساتید مختلفی داشت و به این صورت بود که هر استاد فقط یک مبحث را تدریس میکرد. مباحث در ابتدا بیشتر جنبه آموزشی داشتند، اما رفته رفته این مباحث تدریس شده به صورت کاربردی و ترکیبی در سایر دیگر مباحث تدریس شدند و تمرین هایی برای بررسی میزان یادگیری فرد داده شدند.

پس از انجام دو دوره تمرین، استاد کارآموزی پروژه هایی را به افراد تخصیص داد که قرار شد با راهنمایی ایشان و گزارش دوره ای و جلسات متعدد به انجام برسد. افراد میتوانند به صورت گروهی نیز پروژه را انجام دهند که کیفیت بالاتری مورد انتظار بود. در کنار جلسات رفع اشکالی که استاد برگزار میکردند، افراد گروه برای هماهنگی جلساتی جدا و با هماهنگی خارج از فضای کار برگزار میکردند.

**بنده در این مدت ۴ تمرین و ۱ پروژه به شرح زیر انجام دادم:**

1. تمرین ۱

a. اشکال یابی کد نمایش گراف از اتصالات port ها در فایل های SPICE به صورت نمودار Euler ( زبان Python )

b. تبدیل کد نرم افزار تولید عبارت PDN از فایل SPICE و نمایش Truth Table از C++ به Python

c. تجميع قسمت a و b در یک نرم افزار

2. نوشتن افزونه برای نمایش عکس بر روی صفحه طراحی برای نرم افزار MicroMagic Max

3. افزودن افزونه برای نمایش فایل هایی با فرمت CIF برای نرم افزار MicroMagic Max

4. نوشتن نرم افزار تبدیل فرمت دو نرم افزار Magic و Max به یک دیگر

# فهرست مطالب

2.....	به نام خدا
3.....	با تشکر از فراوان از
4.....	ارزیابی
5.....	چکیده
5.....	فهرست مطالب
9.....	۱. مقدمه
9.....	معرفی حوزه
11.....	موضوع کار آموزی
12.....	۲. تشریح دوره کار آموزی
12.....	دوره آموزشی
13.....	دوشنبه ۲ مرداد
13.....	نحوه کار صفر تا ۱۰۰ یک نرم افزار CAD
13.....	مبانی امنیت در سرویس های وب
14.....	سه شنبه ۳ مرداد
14.....	مبانی مقدماتی Python در محیط Jupyter
14.....	مبانی C و C++
15.....	چهارشنبه ۴ مرداد
15.....	بررسی و مقایسه دو ابزار Sue و XChem
16.....	شنبه ۷ مرداد
16.....	مبانی C و C++
17.....	یکشنبه ۸ مرداد
17.....	مبانی کاربردی Openlane / efabless
17.....	مبانی کاربردی Openlane / efabless
18.....	سه شنبه ۱۰ مرداد
18.....	مبانی مقدماتی Python در محیط Jupyter
18.....	مبانی پیشرفته Python در محیط Jupyter

19.....	دوره عملی .....
19.....	آمادگی .....
19.....	نصب سیستم عامل Ubuntu .....
19.....	نصب مجموعه نرم افزار های Micro Magic .....
20.....	نصب نرم افزار Jade .....
20.....	نصب نرم افزار Magic .....
21.....	تمرین ۱ .....
	۱. اشکال یابی کد نمایش گراف از اتصالات port ها در فایل های SPICE به صورت
21.....	نمودار Euler .....
	۲. تبدیل کد نرم افزار تولید عبارت PUN و PDN از فایل SPICE و نمایش Truth
22.....	Table از C++ به Python .....
23.....	۳. تجمیع قسمت ۱ و ۲ در یک نرم افزار .....
24.....	تمرین ۲ .....
27.....	تمرین ۳ .....
29.....	پروژه .....
29.....	فاز اول: مطالعه و کسب مهارت در نرم افزار های Max و Magic .....
30.....	فاز دوم: بررسی مفاهیم در دو نرم افزار .....
31.....	نام نرم افزار .....
31.....	تکنولوژی مورد استفاده .....
31.....	زمان آخرین تغییر .....
33.....	برچسب (متن) - Label .....
36.....	مساحت پر شده .....
37.....	سلول سازنده - generator cell .....
39.....	طراحی سلسله مراتبی - استفاده مجدد از المان ها .....
41.....	نکات دیگر .....
42.....	فاز سوم: نوشتن برنامه .....
42.....	نحوه کلی کار برنامه .....
43.....	کامپایل برنامه .....
43.....	آموزش کار با برنامه .....
45.....	نمونه خروجی ها :: تبدیل از mag. به max. ....

- 45.....نمونه خروجی ها :: تبدیل از max. به mag.
- 46.....۳. نتیجه گیری
- 47.....۴. پیشنهادات
- 48.....۵. مراجع
- 49.....۶. ضمائم
- 49.....ماتریس تبدیل ۲ بعدی
- 50.....فایل تبدیل لایه ها
- 51.....زبان برنامه نویسی Nim



# ۱. مقدمه

## معرفی حوزه

در دهه 1950 مهندسان برق با داشتن ترانزیستور های کوچک امکان ساخت مدار های پیشرفته تر را پیدا کردند. اما با پیچیده تر شدن مدارها مشکلاتی به وجود آمد. یک مشکل اندازه مدار بود. یک مدار پیچیده مانند کامپیوتر به سرعت وابسته بود. اگر اجزا مدار بزرگ باشد ، سیمهای متصل به آنها باید طولانی باشد. سیگنال های الکتریکی برای عبور از مدار زمان بر بودند ، این اتفاق باعث کاهش سرعت کامپیوتر می شود. این مشکل با اختراع مدار مجتمع حل شد. مدار مجتمع توسط جک کیلبی و رابرت نويس با قطعات و بلوک یکپارچه از مواد نیم رسانا ساخته شد. مدارها را می توان کوچک تر کرد و فرآیند ساخت آن می تواند خودکار باشد.

در اولین تراشه های نیم رسانا دو ترانزیستور قرار داشت. پیشرفت های بعدی باعث شد ترانزیستور های بیشتری ساخته شوند و در نتیجه باعث شد سیستم های بیشتر و پیشرفته تری ساخته شوند. اولین مدارهای مجتمع تنها چند دستگاه را در خود جای می داد تقریباً به اندازه ده دیود و همچنین ساخت یک یا چند دروازه منطقی روی آن امکان پذیر است. در گذشته به عنوان یکپارچه سازی در مقیاس کوچک (SSI) شناخته می شد ، پیشرفت این روش باعث شد تا در آن بتوان دستگاه هایی با صدها دروازه منطقی قرارداد که به عنوان ادغام در مقیاس متوسط (MSI) شناخته می شوند. پیشرفت های بیشتر منجر به ادغام در مقیاس بزرگ (LSI) ، یعنی سیستم هایی با حداقل هزار دروازه منطقی شد. فناوری کنونی از این علامت بسیار دورتر شده است و ریزپردازنده های امروزی دارای میلیون ها دروازه و میلیارد ها ترانزیستور منفرد هستند ( VLSI - very large scale integration ).

امروزه برای هیچکس تعجب آور نیست که تولید مدار های VLSI یک قدرت محسوب میشود. شرکت ها برای تولید مدار های خود از نرم افزار های طراحی VLSI استفاده میکنند. برخی شرکت ها نرم افزار خاص خود را برای این کار مینویسند و برخی از نرم افزار های عموم استفاده میکنند.

گفتنی است که قیم این نرم افزار ها برای استفاده صنعتی بسیار گران است و به طور مثال لایسنس ۱ ساله آن حدود ۱ میلیون دلار تمام میشود.

در این دوره ما با نرم افزار های رایگان طراحی VLSI کار کردیم و سعی کردیم به آن قابلیت هایی اضافه کنیم.

نرم افزار های مورد استفاده به شرح زیر هستند:

- **Micro-Magic Max**: این نرم افزار در سال ۲۰۰۰ نسخه Public Domain ای به همراه

کد های آن به صورت متن باز منتشر کرد. یکی از قدرتمند ترین نرم افزار ها تا در آن زمان در حوزه خود بوده است که امروزه نیز استفاده میشود

- **Magic I**: این نرم افزار که دائما و حتی تا به امروز نیز بروزرسانی میشود، توسط آقای

Tim Edwards که دانش آموخته دانشگاه Standford و Hopkins میباشد طراحی

شده. آقای Edwards زمانی که در شرکت طراحی VLSI به نام MultiGiG فعالیت

میکرده نوشتن این نرم افزار را شروع کرده است. این نرم افزار بزرگترین نرم افزار رایگان

و open source در این حوزه است

به این نرم افزار ها EDA یا Electronic Design Automation و امروزه طراحی IC بدون آن ها غیر

ممکن است. این نرم افزار ها کارهای مختلفی مانند:

1. اشکال یابی طراحی

2. استفاده modular و سلسله مراتبی

3. ایجاد و استفاده component

4. شبیه سازی

و ... را دارا میباشند.

# موضوع کار آموزی

در دوره کار آموزی بیشتر هدف موارد زیر بود:

1. یادگیری کار و طراحی با نرم افزار های EDA

2. اضافه کردن قابلیت به نرم افزار های EDA

گفتنی است که برای رسیدن به این مرحله، نیازمند دانش های زمینه ای بودیم که اکثرا در کلاس های آموزشی پوشش داده شد.

## ۲. تشریح دوره کارآموزی

### دوره آموزشی

دوره آموزشی حدود ۲ هفته بود که شروع آن از تاریخ دو شنبه ۲ مرداد تا جمعه ۱۳ مرداد به طول انجامید.

ابتدای دوره به آموزش و آشنا کردن افراد با تکنولوژی ها و مفاهیم اولیه سپری شد. دوره آموزشی اساتید مختلفی داشت و به این صورت بود که هر استاد فقط یک مبحث را تدریس میکرد. مباحث در ابتدا بیشتر جنبه آموزشی داشتند، اما رفته رفته این مباحث تدریس شده به صورت کاربردی و ترکیبی در سایر دیگر مباحث تدریس شدند و تمرین هایی برای بررسی میزان یادگیری فرد داده شدند.

### نحوه کار صفر تا ۱۰۰ یک نرم افزار CAD

در این کارگاه جناب مهندس وفایی نژاد ابتدا توضیحات کلی در خصوص روند کلی کار دادند و برای یادگیری بهتر، فرآیند را در نرم افزار های Sue ( برای طراحی منطقی ) و HSpice و Vcs ( برای شبیه سازی ) انجام دادند.

### مبانی امنیت در سرویس های وب

در این کارگاه، جناب مهندس زیوری ابتدا به توضیح معماری کلی وب و وبسایت ها و برخی از protocol های ارتباطی شبکه پرداختند، سپس کمی در مورد دیتابیس ها و امنیت و ارتباط با آن ها را توضیح دادند. بعد از آن رصد و دستکاری اطلاعات ارسالی بین مرورگر و server را با ابزار های Monitoring مانند قسمت network در مرورگر و نرم افزار burp suite ، آموزش دادند.

ایشان در ادامه کارگاه حملات امنیتی روی سایت های آسیب پذیر آزمایشی انجام دادند، یکی از مجموعه سایت های مورد استفاده از آدرس <https://github.com/digininja/DVWA> قابل دسترسی است.

حملات صورت گرفته در سطوح مختلف مانند code injection، و دستکاری داده های ارسالی بودند. ایشان همچنین راه های مقابله با اینگونه حملات را شرح دادند.

### مبانی مقدماتی Python در محیط Jupyter

دو کارگاه پشت سر هم که توسط خانم مهندس ایزدی نسب برگزار شد. ایشان ابتدا در مورد زبان برنامه نویسی پایتون و حوزه های کاربرد آن بحث کردند سپس مبانی مقدماتی پایتون را در نرم افزار Jupyter به صورت تعاملی و مرحله به مرحله آموزش دادند. مبانی مقدماتی که ایشان پوشش دادند عبارتند از:

1. تعریف متغیر
2. انواع داده ای در زبان پایتون
3. تابع های اولیه
4. ایجاد حلقه های for و while و استفاده از آن ها
5. تعریف تابع
6. استفاده از ماژول های دیگر ( import کردن )
7. تعریف کلاس

### مبانی C و C++

در جلسه اول این سری کارگاه، دکتر پشت کوهی ابتدا در مورد تاریخچه این دو زبان توضیحاتی را بیان کردند و سپس در مورد کاربرد های این دو مثال هایی ذکر کردند.

ایشان در ادامه به مفهوم static type بودن زبان پرداختند و بر همین روال به توضیح نحوه تعریف متغیر و تابع پرداختند. در ادامه المان های کنترلی if و else بررسی شدند و از موارد تدریس شده مثال حل کردند

### بررسی و مقایسه دو ابزار Sue و XChem

در این کارگاه که توسط جناب آقای مهندس وهاب برگزار شد، ایشان ابتدا به بررسی کاربرد این دو نرم افزار پرداختند. سپس چند نمونه برد منطقی را در هر دو نرم افزار طراحی کردند. در ادامه کار تفاوت میان کارکرد دو نرم افزار XChem و Sue و نحوه ذخیره سازی آن ها بررسی شد.

ایشان در ادامه به extensible بودن دو نرم افزار پرداختند و توضیحات اولیه ای را در مورد چگونگی نوشتن Plug-In برای هردو و زبان مورد استفاده برای این کار را دادند

### اصولی مقدماتی TCL

آقای مهندس وهاب در کارگاه دوم به بررسی مفاهیم و اصول برنامه نویسی در زبان TCL که به دلیل زبان مورد استفاده در نرم افزار های طراحی مورد توجه خاصی قرار دارد، پرداختند. جا انداختن اصول و مقدمات این زبان کار ساده ای نبود زیرا اولاً ساختار آن با زبان هایی که امروزه مورد استفاده قرار میگیرد و ثانیاً منابع آموزشی برای آن نیز کم بود.

ایشان به بررسی مفاهیم زیر در زبان TCL پرداختند:

1. تعریف متغیر
2. تابع های پیشفرض
3. فراخوانی تابع
4. تعریف حلقه های for و while
5. تعریف تابع
6. مفهوم evaluate و تابع eval
7. ساختمان داده های List و Array

### مبانی C و C++

در این کارگاه جناب دکتر پشت کوهی به ادامه مباحث گفته شده پرداختند که شامل تعریف struct ها و class ها میشد. ایشان هم چنین فلسفه الگوی برنامه نویسی به سبک شیء گرایی یا Object Oriented Programming را شرح دادند و مزایای آن را نسبت به الگوی برنامه نویسی رویه ای یا procedural ذکر کردند.

### مبانی کاربردی C و C++

این کارگاه توسط مهندس عالیخانی برگزار شد که دانش آموخته ورودی ۹۸ مهندسی کامپیوتر همین دانشگاه است. ایشان ابتدا برنامه Astran-MS که برای تولید layout مورد استفاده قرار میگرفت را شرح دادند و سپس به شرح پروژه سال گذشته خود که در مورد "تولید خودکار Layout دبه کمک نرم افزار Astran-MS" بود، پرداختند.

پروژه Astran-MS مشکلاتی داشت که کار automation (خودکار سازی) را سخت میکرد. یکی اینکه پروژه محیط Command Line نداشت و فقط محیط گرافیکی داشت، و دیگر اینکه به پروژه به editor کد code::blocks وابستگی شدید داشت.

ایشان شرح دادند که چگونه کد های این نرم افزار را به گونه ای تغییر دهند که اولاً بدون نیاز به نرم افزار code::blocks کامپایل و اجرا شود، و در مرحله دوم قابلیت به آن اضافه کردند که بدون نیاز به اجرا کردن محیط گرافیکی ( GUI ) انجام کار های خودکار سازی ممکن باشد



### مبانی کاربردی Openlane / efabless

دو کارگاه پی در پی توسط خانم مهندس قائم نیا برگزار شد که ایشان ابتدا توضیحاتی در مورد مشکلات طراحی از صفر را بیان کردند و سپس به معرفی و بیان هدف شرکت efabless و پروژه OpenLane پرداختند.

ایشان مفهوم PDK را توضیح دادند و PDK شرکت SkyWater با نام Sky130 را که با همکاری دو شرکت SkyWater و Google ایجاد شده، معرفی کردند. سپس چند پروژه که یکی از آن ها یک PLL بود را با PDK نسخه Sky130 را شبیه سازی کردند و Layout آن ها را در نرم افزار طراحی Magic و KLayout نمایش دادند و اجزای آن را توضیح دادند.

### مبانی کاربردی Openlane / efabless

این کارگاه توسط آقای مهندس انصاری برگزار شد. ایشان بیشتر به مباحث پایه ای و بنیادی مورد نیاز برای فهم اصطلاحات پرداختند. ایشان فرآیند های ساخت یک IC که شامل موارد زیر میشود را به همراه اصطلاحات هر بخش توضیح دادند:

1. Synthesis
  - a. Optimization
  - b. Mapping
2. Floor Planning
3. Placement
4. Routing
5. Photolithography
6. CTS ( Clock Tree Synthesis )

ایشان همچنین به معرفی نرم افزارهای مورد استفاده در این حوزه مانند Magic و KLayout و Yosys نیز پرداختند.

### مبانی مقدماتی Python در محیط Jupyter

ادامه کارگاه های خانم مهندس ایزدی نسب که آخرین بار در روز سه شنبه ۳ مرداد برگزار شده بود. ایشان در این کارگاه در خصوص نوشتن مقاله در Jupyter و مباحثی چون:

1. LaTeX
2. Markdown

پرداختند و نمونه های عملی از این کار را نشان دادند

### مبانی پیشرفته Python در محیط Jupyter

این دو کارگاه در واقع ادامه کارگاه های خانم مهندس ایزدی نسب بودند که توسط آقای مهندس حسین نیا برگزار شدند. ایشان در این کارگاه ها بیشتر از تعامل محیط Jupiter با سایر برنامه ها و نرم افزار ها اشاره کردند و چند نمونه عملی از پروژه های خود را در این باره نشان دادند.

ایشان همچنین به معرفی کتابخانه های رسم نمودار و کار های آماری مانند Pandas و MatPlot در پایتون پرداختند

ایشان سپس درباره تعاملی کردن صفحات Jupyter با افزودن المان های HTML و کد های JavaScript پرداختند و چند نمونه مثال کاربردی به دانشجویان نشان دادند.

### آمادگی

قبل از شروع دوره باید Task های لازم را انجام میدادیم. تمرین های داده شده معمولا به موارد زیر وابسته بودند:

### نصب سیستم عامل Ubuntu

به دلیل واسبستگی تقریبا همه نرم افزار های مورد استفاده، مجبور بودیم تا از یک مشتق Linux استفاده کنیم که بنابر نظر استاد، Ubuntu را نصب کردیم. استاد محیط مجازی را برای نصب پیشنهاد دادند ولی من سیستم عامل را در کنار Windows نصب کردم و از این بابت مشکلی نبود.

### نصب مجموعه نرم افزار های Micro Magic

نرم افزار های طراحی دیجیتال و EDA شرکت Micro Magic در سال ۲۰۰۰ به صورت open source در آمدند که هنوز هم در صنعت استفاده دارند. مراحل نصب این برنامه در سیستم عامل Ubuntu توسط استاد عزیز جناب دکتر غزنوی قوشچی گذاشته شد و در نصب آن مشکلی نبود

## نصب نرم افزار Jade

نرم افزار های EDA تحت وب چند سالی است که جلب توجه کرده اند و مزیت های آن ها برای همه اثبات شده. یکی از این EDA های تحت وب Jade نام دارد که از ما خواسته شد تا آن را نصب کنیم و یک طرح در آن رسم کنیم. آدرس نرم افزار:

<https://github.com/6004x/jade>

## نصب نرم افزار Magic

بنده فایل های binary نرم افزار را پیدا نکردم و مجبور شدم کد آن را دانلود و بعد Build کنم.

در ابتدا هنگام build کردن با اروری مواجه میشدم که با راهنمایی سازنده آن متوجه شدم مشکل از شناسایی نکردن کتابخانه های زبان برنامه نویسی TCL و کتابخانه گرافیکی Tk است.

<https://github.com/RTimothyEdwards/magic/issues/261>

بعد از جستجو در اینترنت و سایت هایی نظیر stack-overflow بالاخره توانستم نرم افزار را Build و اجرا کنم.

# تمرین ۱

تمرین ۱ شامل ۳ قسمت بود که به ترتیب همه را شرح میدهم:

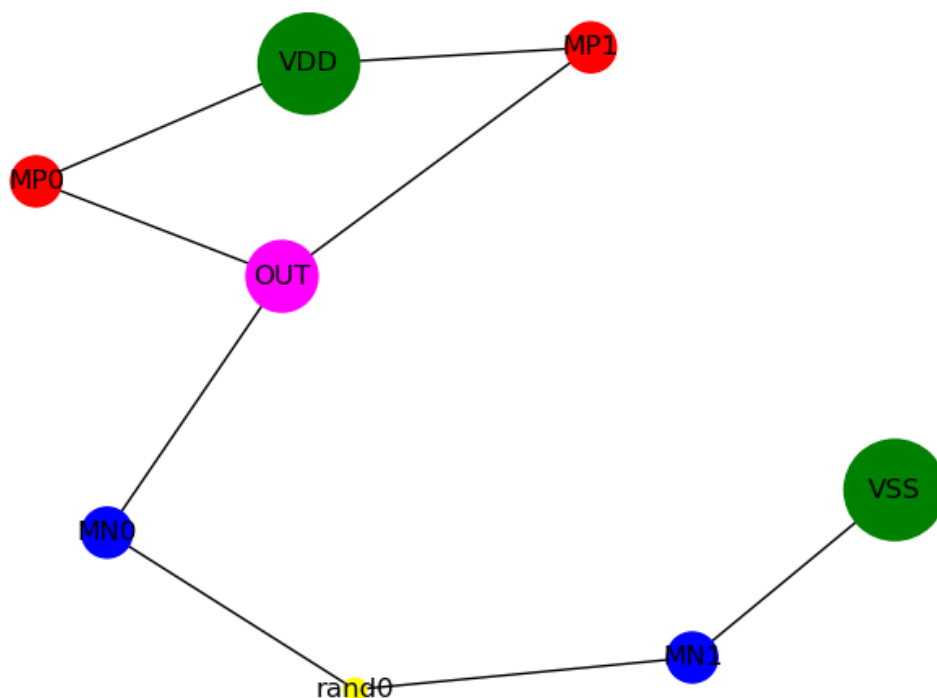
## ۱. اشکال یابی کد نمایش گراف از اتصالات port ها در فایل های SPICE به صورت

### نمودار Euler

زبان این برنامه پایتون بود. در هنگام اجرای آن دچار مشکلاتی شدم که با بررسی هایی که انجام دادم متوجه شدم این برنامه از ورژنی از کتابخانه networkx استفاده میکند که رفتار برخی از تابع های آن امروزه متفاوت است. کتابخانه networkx یک کتابخانه کار با گراف و نمایش آن است که الگوریتم های رایج پیمایش گراف، افزودن یال و گره و... را دارد.

بعد از رفع اشکال، دست به اصلاحاتی در کد زدم و حجم آن را کاهش دادم.

### نمونه خروجی:



## ۲. تبدیل کد نرم افزار تولید عبارت PUN و PDN از فایل SPICE و نمایش Truth Table از C++ به Python

نمونه خروجی برنامه ( سمت چپ ) به ازای فایل nand3.sp ( سمت راست )

```
README.md /home/hamidb80/Documents/1 ...
# LEX - SPICE STD-CELL ARCS EXTRACTOR
PUN Expressions:
OUT=(C*(A*B))

PDN Expressions:
OUT=(C*(A*B))
-----
Expression After Flattening:
!((C*(A*B)))*!((C*(A*B)))
-----
TRUTH TABLE:
000|1
100|1
010|1
110|1
001|1
101|1
011|1
111|0
-----
```

```
nand3.sp
.SUBCKT NAND3D0BWP A B C OUT VDD VSS
MP0 VDD A OUT VDD PCH W=2.5E-06 L=0.04E-06
MP1 VDD B OUT VDD PCH W=2.5E-06 L=0.04E-06
MP2 VDD C OUT VDD PCH W=2.5E-06 L=0.04E-06
MN0 OUT A rand0 VSS NCH W=2.5E-06 L=0.04E-06
MN1 rand0 B rand1 VSS NCH W=2.5E-06 L=0.04E-06
MN2 rand1 C VSS VSS NCH W=2.5E-06 L=0.04E-06

.ENDS
```

در این قسمت چون C++ یک زبان static type است، ابتدا کد های آن را به زبان Nim که  
زبانی static type است و ظاهری شبیه زبان Python دارد، تبدیل کردم و بعد از اینکه از درستی کارکرد  
برنامه تبدیلی مطمئن شدم، آن را دوباره به Python تبدیل کردم.

گفتنی است که مواردی را در کد های C++ هنگام تبدیل به Python رعایت کردم، مثلا تا جایی  
که ممکن بود از توابع پیشفرض خود زبان Python برای کار هایی مثل جستجو در آرایه، Print کردن و  
عملیات های روی string استفاده کردم

### ۳. تجميع قسمت ۱ و ۲ در يك نرم افزار

در اين قسمت توابع و كلاس هاى مشترك بين دو برنامه را پيدا كردم و در يك فايل جدا قرار دادم. بعد براى ايجاد Command Line Interface يك فايل main.py درست كردم كه با آن ميتوانستيد به هردو نرم افزار دسترسى داشته باشيد

گفتنى است كه براى تجميع اين دو برنامه مجبور شدم اصلاحات ديگرى را نيز در كد آن ها انجام دهم مثلاً متغير هاى عمومى را حذف كنم و همه كد هاى كه مستقيم اجرا ميشد را در تابع جداگانه با ورودى هاى مورد نياز قرار دهم.

در نهايت، استفاده برنامه CLI نهايى به صورت زير است:

USAGE:

```
app -lex <PATH_TO_.sp> <OUTPUT_FILE_PATH.sp>
app -sasc <wpwn ratio> <PATH_TO_INPUT.sp> <OUTPUT_FILE_PATH.arcs>
```

...

نشان -lex براى برنامه تمرين ۲ و نشان -sasc براى برنامه تمرين ۱ است

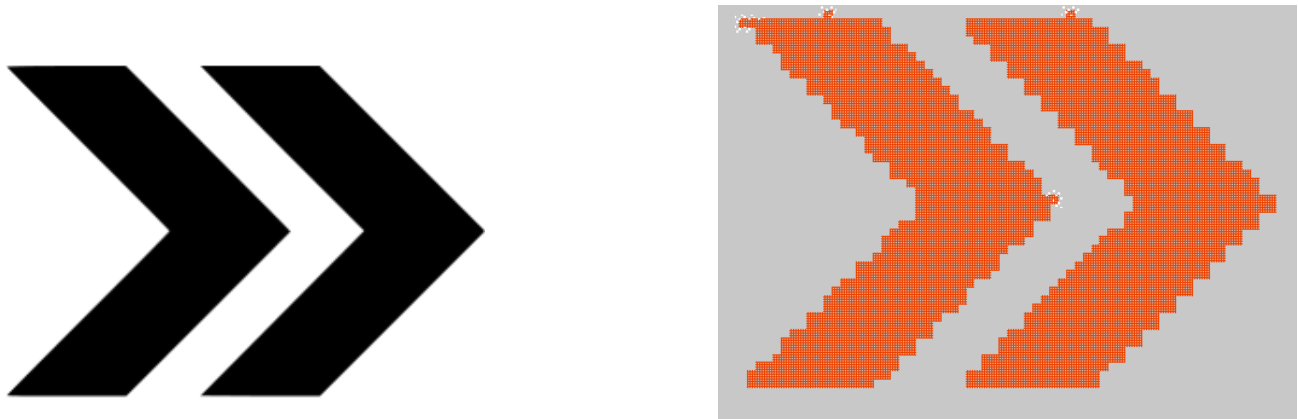
## تمرین ۲

موضوع این تمرین "نوشتن افزونه برای نمایش عکس بر روی صفحه طراحی برای نرم افزار *MicroMagic Max*" بود. شرح تمرین به این صورت بود که یک فایل کد به زبان C در اختیار ما قرار گرفته بود و باید برای آن فقط یک افزونه با زبان TCL مینوشتیم که با کد C و نرم افزار Max ارتباط برقرار کند. کد C مورد نظر در این آدرس موجود است:

<https://github.com/stineje/xchiplogo>

من ابتدا فکر کردم که تنها کار مورد نیاز برای اینکار، قرار دادن تک تک پیکسل های تصویر روی Layout است و تغییر خاصی نیاز نیست، بخاطر همین بجای کد C ای که استاد به من داد که شامل حدودا 560 خط میشد، یک کد به زبان Nim نوشتم که این کار را در فقط ۱۰ خط انجام میداد.

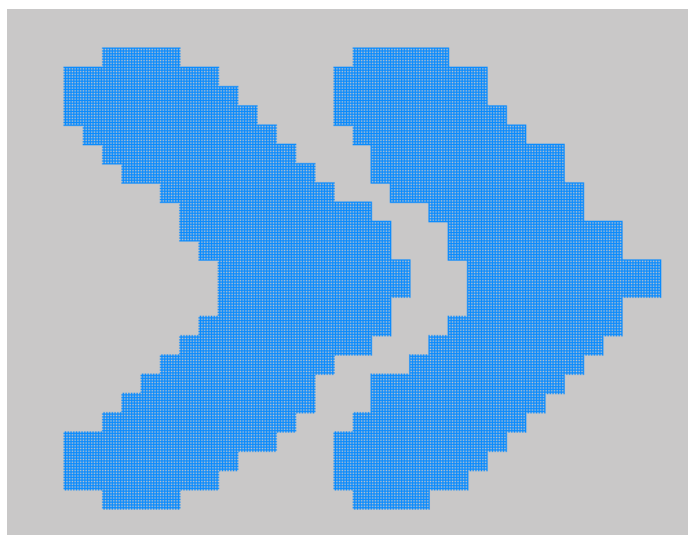
برای تست اولیه این تصویر سمت چپ را به آن دادم. در خروجی تصویر سمت راست را تحویل گرفتم:



همه چیز به نظر خوب بود تا اینکه استاد اشاره یادآوری کردند که خروجی تولیدی شامل ارور های DRC یا Design Rule Check است. این ارور های به معنی این هستند که Layout طراحی شده هنگام طرح ریزی و ساخت به دلیل برخی دلایل (مانند نزدیکی بیش از حد نقاط شکست و تیز) دچار مشکل میشود.

بخاطر همین مجبور شدم به همان کد C برگردم. نتیجه نهایی بدون ارور است:





پنلی که افزونه نوشته شده در اختیار کاربر قرار میدهد تا فایل عکس مورد نظر را انتخاب کند، به شکل زیر است:

Image Paster

/home/hamidb80/Downloads/arrow.png

Open an Image

Layer: m1

Scale: 1

Threshold Before: 8

Threshold After: 4

Image Width: 32

Image Height: 32

Draw

متغیر های Threshold After و Threshold Before برای نرم کردن گوشه های تیز تصویر است که با آزمون و خطا میتوان مقدار مناسب آن ها را که از 4 تا 16 قابل تغییر است، پیدا نمود.

نکته ای که در مورد این تمرین وجود دارد این است که کد C فقط ورودی تصویر به فرمت pbm میگیرد. افزونه نوشته شده برای رفع این مشکل ابتدا تصویر انتخاب شده را با نرم افزار Image Magick که معمولا در سیستم عامل های Linux به صورت پیش فرض نصب است، به این فرمت تبدیل میکند و سپس به این نرم افزار میدهد.

بنده در قسمت اول در کد Nim خود ابتدا کتابخانه ای برای خواندن فایل های pbm پیدا نکردم، پس کتابخانه ای به نام fastpnm نوشتم که این کار را بکند.

اما بعد از فهمیدن اینکه نرم افزار در قسمت اول درست کار نمیکند، از کار خود پشیمان شدم. کد کتابخانه در آدرس زیر قابل دسترسی است:

<https://github.com/hamidb80/fastpnm>

البته در هنگام تحقیق در مورد فرمت pbm. به چیز های جالبی رسیدم مثلا اینکه این فرمت از مجموعه فرمت pnm است که مجموعا شامل ۳ فرمت است و به دلیل اینکه سرور های ایمیل قدیمی بعضا امکان ارسال فایل های binary را نداشتند، این تصویر هارا میفرستادند که به صورت ASCII ذخیره میشد. این ۳ فرمت عبارتند از:

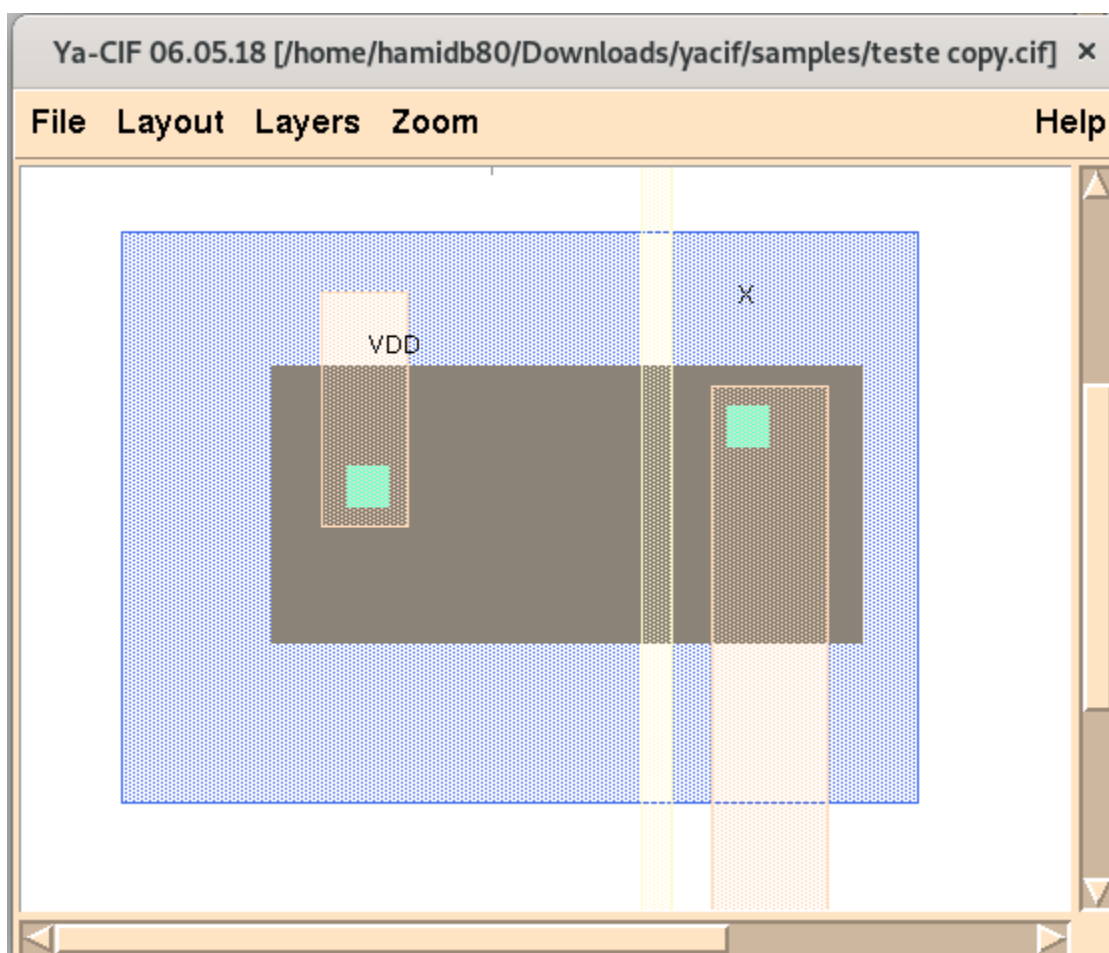
1. Pbm: پیکسل ها به دو رنگ سیاه یا سفید ذخیره میشوند
2. Ppm: پیکسل ها میتواند مقادیر طوسی رنگ روشن و تیره داشته باشند
3. Pgm: پیکسل ها به صورت تماما رنگی ذخیره میشوند

## تمرین ۳

موضوع این تمرین "افزودن افزونه برای نمایش فایل هایی با فرمت CIF برای نرم افزار MicroMagic Max" بود.

شرح پروژه به این صورت بود که قبلا برنامه ای با نام yaCIF یا Yet Another CIF Viewer به زبان TCL نوشته شده بود و فقط لازم بود کد های TCL آن را طوری تغییر دهیم که در کنار نرم افزار Max و به صورت یک افزونه اجرا شود.

نتیجه نهایی به صورت زیر است:



بزرگترین چالش این پروژه اصلاح کد های آن به صورتی است که با نرم افزار Max تداخل نداشته باشد. از آن جایی که زبان TCL مفهومی به نام name space یا module ندارد، برای کار نکردن نرم افزار تنها کافی است نام یک متغیر عمومی یا تابع بین دو کد مشترک باشد، در این صورت انتفاقات عجیبی ممکن است رخ دهد

چون در تمرین ۲ مهارت های نوشتن برنامه های Tk و TCL را کسب کرده بودم، انجام این تمرین خیلی مشکل نبود.

البته برای اشکال یابی برنامه مجبور شدم توضیحات مربوط به فایل های CIF. را مطالعه کنم:

[https://en.wikipedia.org/wiki/Caltech\\_Intermediate\\_Form](https://en.wikipedia.org/wiki/Caltech_Intermediate_Form)

موضوع پروژه "نوشتن نرم افزار تبدیل فرمت دو نرم افزار *Magic* و *Max* به یک دیگر" بود.

### فاز اول: مطالعه و کسب مهارت در نرم افزار های *Magic* و *Max*

برای اینکه بتوانم فرمت این دو نرم افزار را بهم تبدیل کنم، اول باید هر دو نرم افزار را خیلی خوب بشناسم. پس فاز اول پروژه را به یادگیری این دو به صورت خود آموز و از روی *Manual* و *Tutorial* های آن ها اختصاص دادم.

راهنمای نرم افزار *Max* در پوشه آن و راهنمای نرم افزار *Magic* در آدرس زیر موجود است:

<http://opencircuitdesign.com/magic/>

در این مدت سعی کردم ابتدا راهنمای مورد نظر را خوانده و سپس موارد ذکر شده را به صورت عملی انجام دهم.

همچنین از ویدئو های آموزشی برای نرم افزار *Magic* در Youtube کمک گرفتم.

## فاز دوم: بررسی مفاهیم در دو نرم افزار

بعد از خواندن راهنمای دو نرم افزار و کسب دانش اولیه، اقدام به تحلیل فایل های ذخیره سازی آن ها کردم. چیزی که نظرم را جلب کرد این بود که ساختار فایل های ذخیره شده خیلی باهم تفاوت نداشتند و همه مفاهیم به جز generator cell در نرم افزار Max ، در Magic هم موجود بودند و برعکس.

در هر مفهوم را در دو نرم افزار بررسی میکنم:

**توجه:** در همه تصاویر مقایسه ای قرار داده شده در این قسمت، سمت چپ متعلق به نرم افزار Max و سمت راست برای نرم افزار Magic میباشد

## نام نرم افزار

در اولین خط فایل های ذخیره شده، نام نرم افزار نوشته میشود. در نرم افزار Magic فقط نام magic نوشته شده اما در نرم افزار Max نام نرم افزار به همراه ورژن آن نوشته میشود. به طور مثال خط زیر بیان میکند که فایل مورد نظر با ورژن ۳ نرم افزار Max تولید شده:

max 3

مثال:

max 3

magic

## تکنولوژی مورد استفاده

در ابتدای هر دو فرمت ذخیره سازی، الگوی زیر دیده میشود که <tech\_name> نام تکنولوژی مورد استفاده است:

tech <tech\_name>

مثال:

tech mmi25

tech scmos

## زمان آخرین تغییر

زمان در هر دو نرم افزار به صورت unixtime ذخیره میشود که یک عدد صحیح است. در نرم افزار Magic تنها با الگوی زیر زمان آخرین تغییر قابل فهمیدن است:

timestamp <unixtime>

مثال:

timestamp 500618121

اما در نرم افزار Max این الگو یک مقدار پیچیده تر است. اطلاعات زمانی و تعداد ویرایش ها در بلوکی به نام VERSIONS ذخیره میشوند. عدد اول زمان برحسب unixtime است و عدد دوم نشان دهنده تعداد ویرایش ها روی فایل است:

vMAIN <unitttime> <revision\_number>

مثال:

```
SECTION VERSIONS {  
vMAIN 1692631583 1  
vDRC 1692631583 1  
vBBOX 1692631583 1  
} SECTION VERSIONS
```



## برچسب (متن) - Label

برچسب ها در هر دو نرم افزار به یک الگو ذخیره میشوند و تفاوت در آن ها بسیار جزئی است. در نرم افزار Max این اطلاعات درون بلوک LABELS و در نرم افزار Magic این اطلاعات در بلوک labels قرار دارند.

```
SECTION LABELS {  
lab m2 6 15 6 20 0 2 Vdd!  
lab m2 6 15 6 20 0 2 Vdd!  
lab m2 -1 20 1 20 0 2 phibar  
lab m2 -1 -10 1 -10 0 2 phi  
} SECTION LABELS
```

```
<< labels >>  
rlabel m2 6 15 6 20 2 Vdd!  
rlabel m2 6 15 6 20 2 Vdd!  
rlabel m2 -1 20 1 20 2 phibar  
rlabel m2 -1 -10 1 -10 2 phi  
<< end >>
```

برچسب ها در نرم افزار Max در قسمتی به نام LABELS قرار دارند و به الگوی زیر هستند:

lab <layer> <x1> <y1> <x2> <y2> <kind> <text>

ذخیره میشود که <layer> نام لایه ای است که آن برچسب به آن تعلق دارد و x1 y2 x2 y2 مختصات آن برچسب هستند. مقدار kind میتواند عددی بین ۰ تا ۶ باید که هرکدام به معنی های زیر است: ( این اطلاعات از فایل های کد برنامه Max استخراج شده )

```
/* Kinds of Labels */
#define LAB_COMMENT      0
/* hidden labels not normally displayed */
#define LAB_HIDDEN      1
#define LAB_LOCAL       2
#define LAB_GLOBAL      3
/* ports */
#define LAB_INPUT       4
#define LAB_OUTPUT      5
#define LAB_INOUT       6
#define LAB_MAX_KIND    6
```

الگوی ذخیره سازی برچسب در نرم افزار Magic به شرح زیر است:

`rlabel <layer> <x1> <y1> <x2> <y2> <orient> <text>`

تفاوت آشکاری که بین این دو وجود دارد، قسمت kind در Max و قسمت orient در Magic است.

در Magic مقدار Orient میتواند عددی بین ۰ تا ۷ باشد که محل و جهت قرار گیری آن را مشخص میکند. هرکدام به معنی زیر است:

```
enum def_orient {  
    ORIENT_NORTH, // 0  
    ORIENT_SOUTH, // 1  
    ORIENT_EAST, // 2  
    ORIENT_WEST, // 3  
    ORIENT_FLIPPED_NORTH, // 4  
    ORIENT_FLIPPED_SOUTH, // 5  
    ORIENT_FLIPPED_EAST, // 6  
    ORIENT_FLIPPED_WEST // 7  
};
```

## مساحت پر شده

```
SECTION RECTS {
layer poly
-1 15 1 20
-1 9 1 11
-1 -1 1 1
-1 -10 1 -5

layer ndif
-2 -5 -1 -1
1 -5 2 -1

} SECTION RECTS
```

```
<< polysilicon >>
rect -1 15 1 20
rect -1 9 1 11
rect -1 -1 1 1
rect -1 -10 1 -5

<< ndiffusion >>
rect -2 -5 -1 -1
rect 1 -5 2 -1
```

در نرم افزار Max مساحت های پر شده درون بلوک RECTS قرار میگیرند. و الگوی زیر را دارند:

```
layer <layer>
<x1> <y1> <x2> <y2>
<x1> <y1> <x2> <y2>
...
```

که <x1> <y1> <x2> <y2> مساحت آن قسمت و <layer> لایه ای که آن مساحت در آن قرار دارد را مشخص میکنند.

در نرم افزار Magic الگو این چنین است

```
<< [layer] >>
<x1> <y1> <x2> <y2>
<x1> <y1> <x2> <y2>
...
```

که [layer] نام لایه را مشخص میکند.

## سلول سازنده - generator cell

در نرم افزار Max مفهومی به نام "سلول سازنده" وجود دارد که واقعیت این است که یک نوع المان پویا است که مثلا میتواند تعداد port ها و مشخصات دیگری که برای آن تعریف شده را تغییر داد.

در صورت ایجاد یک المان با "سلول سازنده" نرم افزار یک نسخه از آن را با ثبت تنظیمات برای جلوگیری از تکرار در فایل Max ذخیره میکند. در استفاده های مجدد از این المان، تنها نام آن با مختصاتی جدید دوباره استفاده میشود.

محتویات یک سلول سازنده مانند یک فایل max. مییاشد که میتوان بلوک های RECTS و LABELS و... داشته باشد

در تکه کد زیر یک المان via با نوع ct\_poly ایجاد شده است:

```
DEF #via!-type!ct_poly!-_version!360385828 "via (S)" "ct_poly"

SECTION RECTS {
layer poly
-290 -290 290 290
layer ct
-150 -150 150 150
layer m1
-240 -240 240 240
} SECTION RECTS
```

نحوه استفاده از آن در Layout به این صورت است که باید در بلوک INSTANCES نوشته شوند. در خط اول نام المان به همراه تنظیمات و ورژن ( که معمولا زمانی برحسب unixtime است ) وجود دارد. در قسمت bbox مساحتی که در صفحه اشغال میکند ( نسبت به مرکز مختصات یعنی نقطه ۰ و ۰، و در قسمت uses محل کپی ها آن مشاهده میشود. یک نمونه مثال:

```
SECTION INSTANCES {
gcell 360385828 #via!-type!ct_poly
bbox -290 -290 290 290
uses {
    _740 1 0 2620 0 1 510
    _0 1 0 -1750 0 1 520
}
} SECTION INSTANCES
```

الگو به صورت زیر است:

```
gcell <version>
#<name>!--<property1>!--<value1>!--<property2>!--<value2>...
bbox <x1> <y1> <x2> <y2>
uses {
    _<id1> <a> <b> <c> <d> <e> <f>
    _<id2> <a> <b> <c> <d> <e> <f>
    ...
}
```

مقدار های <id1> و <id2> نام های دلخواه و متمایزی هستند که نرم افزار Max برای شناسایی آن ها استفاده میکند. مقدار های a b c d e f در واقع مقدار های ماتریس تبدیل ۲ بعدی هستند. که در قسمت ضمائم به آن پرداخته ام.

در تبدیل این مفهوم به نرم افزار Magic، سلول سازنده به عنوان یک فایل جدا در کنار بقیه فایل ها ذخیره میشود. که در قسمت بعد ( طراحی سلسله مراتبی ) بیشتر توضیح داده شده.

## طراحی سلسله مراتبی - استفاده مجدد از المان ها

```
SECTION INSTANCES {
def tut4y
vMAIN 0 0
vDRC 0 0
vBBOX 0 0
bbox -3 -10 210 37
uses {
    tut4y_0 1 0 -68 0 1 -59
    tut4y_1 1 0 -68 0 -1 -79
}
} SECTION INSTANCES
```

```
use tut4y tut4y_0
timestamp 500618087
transform 1 0 -68 0 1 -59
box -3 -10 210 37

use tut4y tut4y_1
timestamp 500618087
transform 1 0 -68 0 -1 -79
box -3 -10 210 37
```

نرم افزار Max برای استفاده از المان های از قبل طراحی شده، الگوی زیر را در بلوک INSTANCES مینویسد:

```
def <max_file_name>
vMAIN 0 0
vDRC 0 0
vBBOX 0 0
bbox <x1> <y1> <x2> <y2>
uses {
    _<id1> <a> <b> <c> <d> <e> <f>
    _<id2> <a> <b> <c> <d> <e> <f>
    ...
}
```

که <max\_file\_name> نام فایل max ای است که از المان آن استفاده میشود. بقیه که دقیقا همان اطلاعاتی است که درقسمت قبل ( سلول سازنده ) ذکر کرده ام.

نرم افزار Magic برای هر استفاده از المان مورد نظر از الگوی زیر استفاده میکند:

```
use <mag_file_name> <id>  
timestamp <unixtime>  
transform <a> <b> <c> <d> <e> <f>  
box <x1> <y1> <x2> <y2>
```

مقدار <mag\_file\_name> نام فایلی طراحی است ( بدون پسوند .mag ) که قرار است از آن استفاده شود.

مقدار <id> همان نام متمایزی است که نرم افزار Magic به آن اختصاص میدهد و او را با این نام میشناسد.

مقدار <unixtime> دقیقاً همان زمانی است که در آن فایل در جلوی قسمت timestamp ذخیره شده.

عبارت box و مقدار <y2> <x2> <y1> <x1> مساحت المان مورد استفاده را نسبت به مبدأ مختصات نشان میدهد که بعداً با تبدیل ماتریسی به مکان دلخواه منتقل میشود.



1. در آخرین فایل های mag. الگوی زیر قرار دارد که نشان دهنده پایان فایل است:

```
<< end >>
```

2. در فایل های max. برای نوشتن comment میتوانیم از # در اول خط استفاده کنیم، مانند

```
# hey this is a comment!
```

3. در فایل های max. قبل از شروع تعریف محتویات المان اصلی و بعد از تعاریف سلول های سازنده، الگوی زیر مشاهده میشود:

```
DEF
```

4. در فایل های max. مفهومی به نام resolution وجود دارد که بیان میکند مختصات های ذکر شده، با چه مقایسی باید نمایش داده شوند. مقدار پیشفرض آن برابر با 0.001 است

```
resolution <scale>
```

## فاز سوم: نوشتن برنامه

به نظر میرسید که فایل های ذخیره سازی دو نرم افزار را با دقت خوبی تحلیل کرده ام. پس شروع به نوشتن نسخه های اولیه کردم. در هر قسمت سعی میکردم خروجی دریافتی را تست کنم تا مطمئن شوم روند کار درست است.

### نحوه کلی کار برنامه

کاربر فایل هایی که میخواد تبدیل کند و پوشه هایی که فایل های استفاده شده در آن ها قرار دارند، به همراه پوشه محل خروجی و فایل "تبدیل لایه ها" و تکنولوژی مقصد به برنامه معرفی میکند.

برنامه همه فایل های معرفی شده به همراه فایل های وابسته را میخواند و با توجه به تغییرات نام لایه ها در فایل "تبدیل لایه ها" و مقایسه هایی که در قسمت های قبل انجام شد، به فرمت مورد نظر تبدیل میکند و در نهایت در پوشه معرفی شده با تکنولوژی خواسته شده ذخیره میکند.

در مورد فایل "تبدیل لایه ها" در ضمائم بخوانید

## کامپایل برنامه

برای کامپایل برنامه باید ابتدا از داشتن کامپایلر زبان برنامه نویسی Nim ورژن ۲ مطمئن شوید. اگر ندارید، از آدرس زیر دانلود کنید:

<https://nim-lang.org/>

سپس در پوشه برنامه این دستور را وارد کنید، فایل خروجی در پوشه bin تولید خواهد شد:

```
nimble gen
```

## آموزش کار با برنامه

برنامه ظاهر CLI دارد، این یعنی اینکه باید آن را در محیط Terminal اجرا کنید. اگر برنامه را بدون هیچ ورودی اجرا کنید، راهنمای برنامه نمایان میشود:

USAGE:

```
./app
```

```
-I <MAX_or_MAG_FILE_1> <MAX_or_MAG_FILE_2> ...
```

```
-O <DEST_DIR>
```

```
-S <SEARCH_DIR_1> <SEARCH_DIR_2> ...
```

```
-T <TECH>
```

```
-L <LAYER_MAPPER_FILE_PATH>
```

برای مثال من در قطع کد زیر از برنامه میخوام فایل ./temp/tut4a.mag را به همراه وابستگی هایش که در پوشه ./temp موجود هستند، با تکنولوژی مقصد mmi25 با تبدیل لایه ای موجود در فایل ./layers.mag2max.cfg تبدیل کند و در پوشه ./out ذخیره کند:

```
./bin/maxmag -I ./temp/tut4a.mag -S ./temp -T mmi25 -L  
./layers.mag2max.cfg -O ./out
```

خروجی برنامه در terminal به صورت زیر است:

```
Conversion Direction: mag => max  
Converting ...  
Done!
```

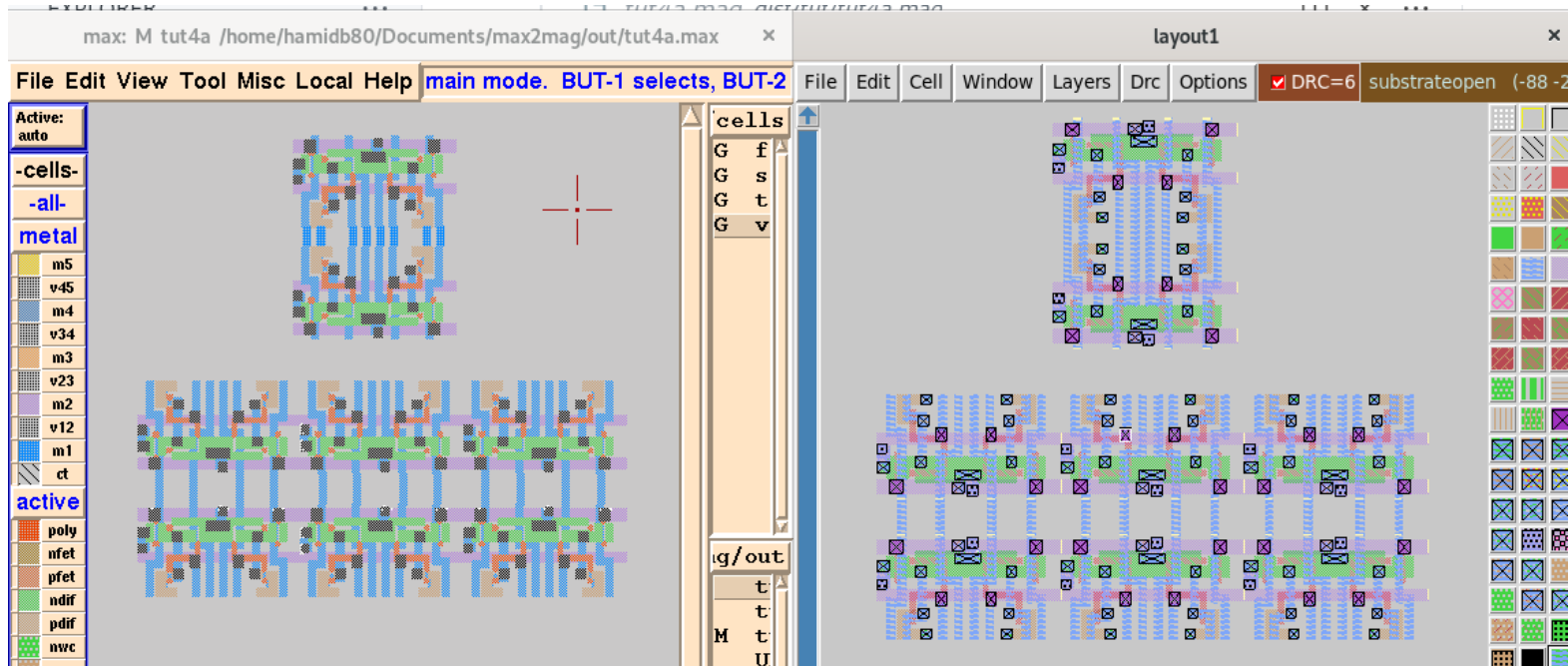
خط اول یعنی اینکه تشخیص داده می‌خواهیم فایل های mag را به max تبدیل کنیم.  
خط دوم قبل از شروع عملیات تبدیل نمایش داده میشود.  
خط آخر Done! نشان دهنده موفقیت آمیز بودن عملیات است

اگر به پوشه out که می‌روم ۳ فایل

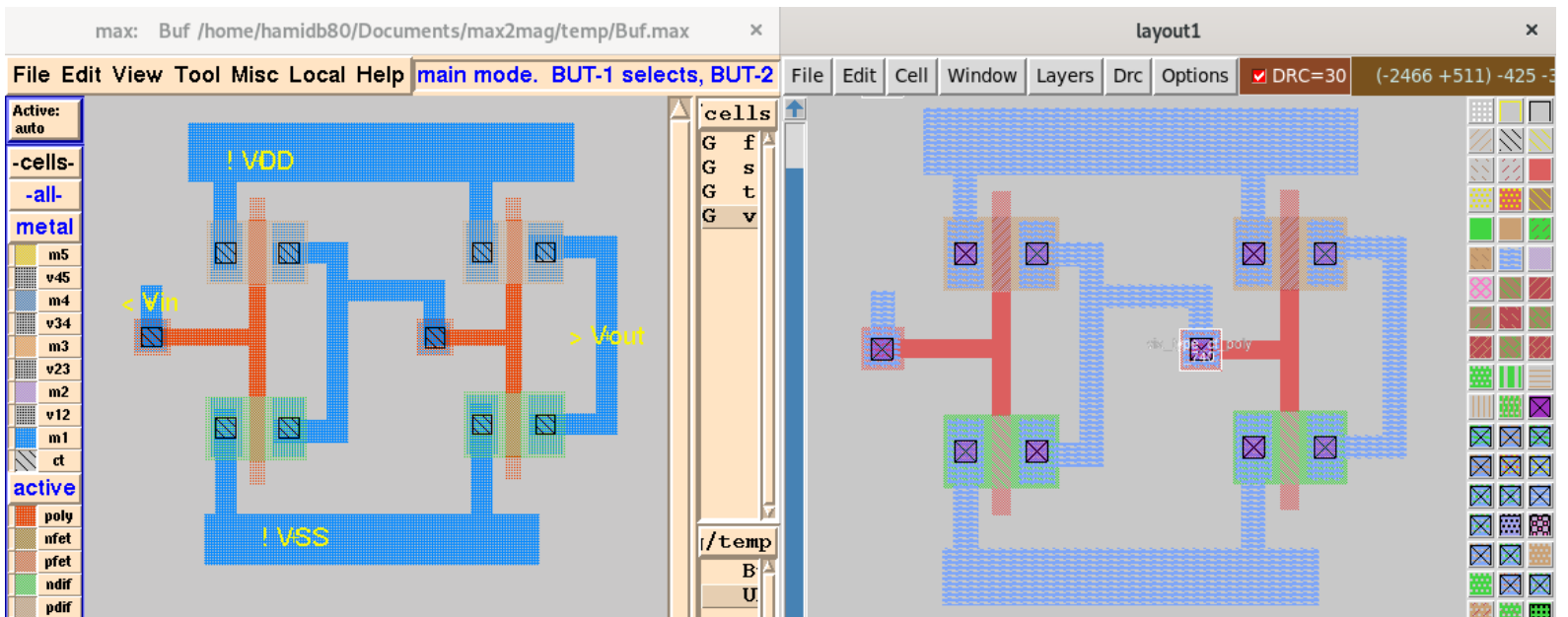
1. tut4a.max
2. tut4x.max
3. tut4y.max

را می‌بینیم. این به این معنی است که نرم افزار وابستگی ها را شناسایی کرده است زیرا tut4a به tut4x و tut4y وابسته است.

نمونه خروجی ها :: تبدیل از .mag به .max



نمونه خروجی ها :: تبدیل از .max به .mag



### ۳. نتیجه گیری

با وجود اساتید صبور و باانگیزه، و البته صبر و حوصله استاد کارآموزی جناب دکتر غزنوی، و برگزاری جلسات رفع اشکال متعدد توسط ایشان، این دوره برای من فرصت خیلی خوبی برای یادگیری مهارت های ذکر شده بود.

قبل از دوره من شناخت بسیار بسیار کمی از این حوزه داشتم و به هیچ وجه از فرصت ها و چالش های این حوزه خبر نداشتم و به گمانم صنعتی را که از تکنولوژی ها حداکثر ۱۰ سال قبل استفاده میکنند، اما حالا این صنعت را صنعتی پویا و دارای فرصت های متعدد جهت پیشرفت و کسب درآمد یافتم.

با تمرین با نرم افزار های EDA ذکر شده و دنبال کردن آموزش های آن به صورت خود آموز، و از همه مهم تر کمک از اینترنت، توانستم کار با آن ها را یادگیرم

اما از همه مهمتر، در قسمت هایی از کار مجبور به خواندن کد های نرم افزار ها شدم که این خود دید من را نسبت به مهندسی نرم افزار باز کرد.

با نوشتن extension یا افزونه برای این نرم افزار ها، و کار کردن با فریمورک گرافیکی Tk و زبان TCL و Python و C++، مهارت نوشتن این چنین نرم افزار ها را کسب کردم. ( ۳ تمرین )

در ادامه در پروژه با نوشتن تبدیل فرمت دو نرم افزار، از معماری ذخیره سازی این دو آگاهی یافتم. در اینجا دانسته های تئوری خود را مانند الگوریتم های گراف و نوشتن parser به عرصه عمل گذاشتم.

البته سوالاتی برای من در این مرحله پیدا شد مانند چگونگی و نحوه کار فایل های technology که عاقبت کار مجبور شدم برای یافتن سوال آن از سازنده نرم افزار و در ادامه در کتاب های الکترونیک جستجو کنم.

## ۴. پیشنهادات

به نظرم دوره عالی بود و چون پروژه ای که توسط استاد به کارآموزان سپرده شده بود، قبلاً انجام نشده بود و جدید بود، از هیجان و استرس خاصی به طور همزمان برخوردار بود، که این واقعا نکته مثبتی است

در کنار دیگر خوبی ها و البته صبور بودن اساتید، به نظر بنده اگر دوره زودتر شروع شود و تمرین های بیشتری در کنار کلاس های آموزشی داده شود، بهتر است. البته باید محدود بودن زمان را نیز در نظر داشت

## ۵. مراجع

- تاریخچه حوزه VLSI:

[https://en.wikipedia.org/wiki/Very\\_Large\\_Scale\\_Integration](https://en.wikipedia.org/wiki/Very_Large_Scale_Integration)

- بیوگرافی سازنده نرم افزار Magic، آقای Tim Edwards:

<http://opencircuitdesign.com/~tim/>

- کتاب CMOS VLSI Design: A Circuits and Systems Perspective

[http://fa.ee.sut.ac.ir/Downloads/AcademicStaff/24/Courses/11/CMOS%20VLSI%20Design%20A%20Circuits%20and%20Systems%20Perspective,%204th%20Edition%20\(2011\).pdf](http://fa.ee.sut.ac.ir/Downloads/AcademicStaff/24/Courses/11/CMOS%20VLSI%20Design%20A%20Circuits%20and%20Systems%20Perspective,%204th%20Edition%20(2011).pdf)

- آموزش سریع Magic

[https://www.eecg.toronto.edu/~jzhu/courses/ece435/magic\\_tutorial.pdf](https://www.eecg.toronto.edu/~jzhu/courses/ece435/magic_tutorial.pdf)

- راهنمای زبان برنامه نویسی TCL

<https://wiki.tcl-lang.org/>

- کتاب Principles of interactive computer graphics

[https://arato.inf.unideb.hu/fazekas.gabor/oktatas/multimedia/BOOKS/newman\\_sprout\\_princ\\_comp\\_graph.pdf](https://arato.inf.unideb.hu/fazekas.gabor/oktatas/multimedia/BOOKS/newman_sprout_princ_comp_graph.pdf)

- توضیحاتی در مورد فایل های technology

<http://rfic.eecs.berkeley.edu/~niknejad/doc-05-26-02/techfile.html>



## ۶. ضمائم

### ماتریس تبدیل ۲ بعدی

این ماتریس قابلیت تبدیل های انتقال، چرخش و تغییر مقیاس را دارا می باشد. عملکرد این ماتریس پایه ریاضی دارد و در کتاب Principles of interactive computer graphics توضیح داده شده:

#### 4-3 MATRIX REPRESENTATIONS

Two-dimensional transformations can be represented in a uniform way by a  $3 \times 3$  matrix (see Appendix I for a discussion of matrix techniques). The transformation of a point  $(x, y)$  to a new point  $(x', y')$  by means of any sequence of translations, rotations, and scalings is then represented as

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{bmatrix} \quad (4-7)$$

#### 58 BASIC CONCEPTS

##### Matrix Formulation of Transformations

The parameters of the  $3 \times 3$  transformation matrix can be arranged to make the matrix represent the simple transformations of translation, rotation, and scaling:

$$\text{Translation: } \begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \quad (4-8)$$

$$\text{Rotation: } \begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-9)$$

$$\text{Scaling: } \begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-10)$$

## فایل تبدیل لایه ها

با توجه به متفاوت بودن نام لایه ها در فایل های تکنولوژی معرفی شده به دو نرم افزار، برای تبدیل درست، فایلی با الگوی زیر ایجاد کنید:

```
<layer_a>  =>  <layer_1>  
<layer_b>  =>  <layer_2>  
...
```

مثال:

 layers.mag2max.cfg layers.mag2max.cfg

```
magic => max
```

```
poly => poly
```

```
pdiffusion => pdif
```

```
ndiffusion => ndif
```

```
diffusion => diff
```

```
ct => pcontact
```

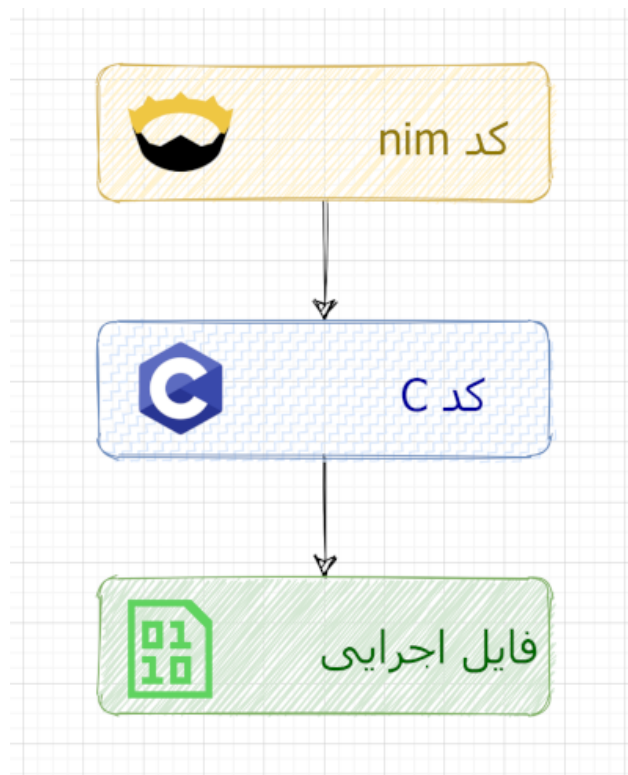
```
polysilicon => poly
```

```
ndiffusion => ndif
```

# زبان برنامه نویسی Nim

زبان برنامه نویسی Nim در سال ۲۰۰۸ توسط آقای دانشجوی دکترای دانشگاه آلمان آقای Andres Rumpf نوشته شد. این زبان که نوشتاری شبیه زبان Python دارد، سعی میکند خوانایی و سرعت را همزمان به ارمغان بیاورد.

کامپایلر این زبان ابتدا کد میانی به زبان های C یا C++ یا JS تولید میکند، سپس کد میانی کامپایل یا اجرا میشود.



وبسایت:

<https://nim-lang.org/>