چیدمان بهینه‌ی ترانزیستورها روی چیپ

با استفاده از الگوریتم ژنتیک

تکلیف پیاده سازی دوم

- ۶ و ۱۲ آذر

مدرس:
دکتر شمس‌فرد
دستیاران آموزشی:
آرش اعتماد
صالح جعفریزاده
یاسمن روحانی‌فر

مقدمه

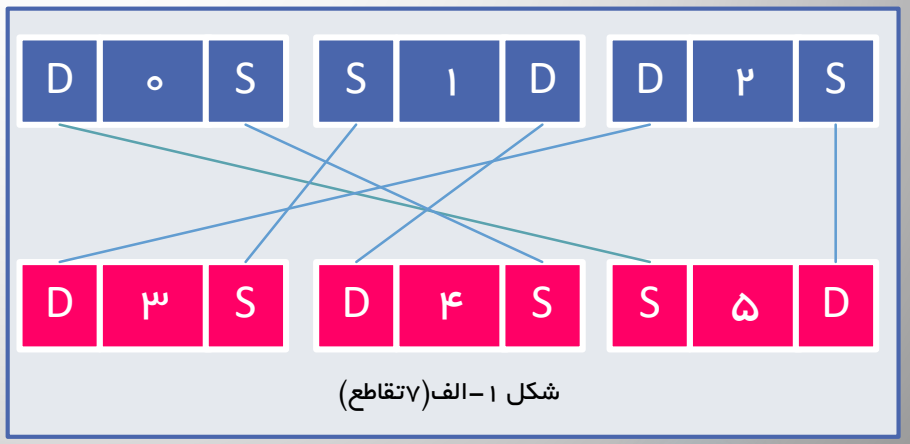
- VLSI(یکپارچه‌سازی کلان‌مقیاس) هنرطراحی مدارهای مجتمع با ترکیب هزاران ترانزیستور رو یک چیپ است.مطالعه بیشتر
- ترکیب ترانزیستورها به معنی برقراری اتصال بین آنها روی چیپ، برای به‌وجود آوردن کاربردی خاص است و طبیعتا ترکیب‌های مختلف بازدهی و هزینه‌ی طراحی و ساخت متفاوتی دارند.
- در این تکلیف یک زیر مسأله کوچک از مسأله‌ی طراحی VLSIمطرح می‌شود. زیرا که ارائه راهکاری بهینه برای طراحی VLSI فراتر از یک تکلیف است.کتاب الگوریتم ژنتیک برای طراحی VLSI
- مساله مطرح شده در این تکلیف چیدمان ترانزیستورهای CMOS روی مدار به‌گونه‌ای است که اتصال‌های آنها کمترین تداخل را باهم داشته باشند.

مفاهیم (مواردی که مطرح می‌شود برای تعریف مسأله بوده و ساده‌شده‌ی مفاهیم مربوط به علم طراحی مدارهای مجتمع کلان مقیاس می‌باشد)

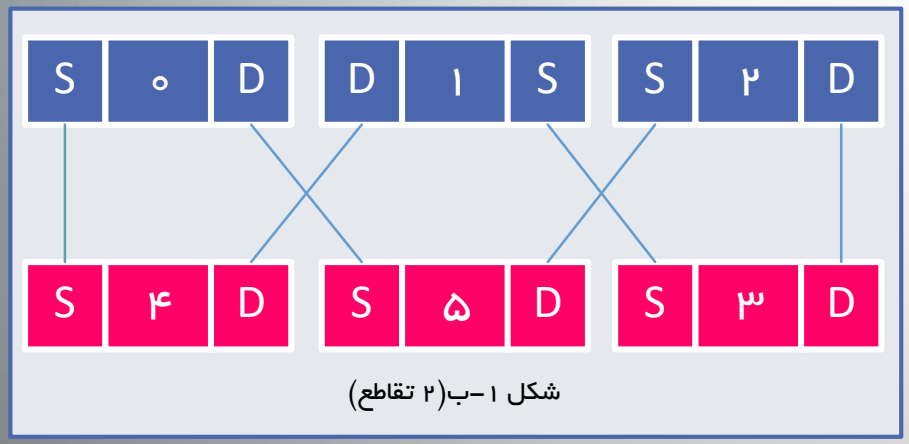
- ترانزیستور:
  - قطعاتی الکترونیکی برای کنترل جریان در مدار می‌باشند و دارای دو پایه Source و Drain هستند.
  - به دو دسته PMOS و NMOS تقسیم می‌شوند.
  - یک مدار از اتصال پایه‌های ترانزیستورها به هم ایجاد می‌شود.
  - برای اتصال یک پایه به پایه دیگر سیمی به صورت خط مستقیم بین آنها قرار می‌گیرد.
- چیپ:
  - یک چیپ از دو ردیف ترانزیستور تشکیل می‌شود.
  - ترانزیستورهای NMOS در ردیف بالا و ترانزیستورهای PMOS در ردیف پایین قرار می‌گیرند.
- تقاطع:
  - در صورتی که سیمی از روی سیم دیگری بگذرد یک تقاطع رخ داده است.

مسأله

- در ابتدا اطلاعات تمام ترانزیستورها (شامل نوع و پایه‌های ترانزیستورهایی که قرار است به آنها متصل شوند)به شما داده می‌شود.
- شما باید با استفاده از الگوریتم ژنتیک ترتیب چیدمانی برای این ترانزیستورها در ردیف‌های خود روی چیپ بیابید که بعد از اتصال ترانزیستورها به هم کمترین تقاطع ایجاد شود.
  - (دقت کنید که هر ترانزیستور فقط میتواند در ردیف خود جابجا شود و جهت قرار دادن ترانزیستورها هم برای رسیدن به حالت بهتر میتواند عوض شود)
- برای مثال اطلاعات چیدمانِ نشان‌داده شده در شکل ۱ –الف به شما داده‌می‌شود و در صورتی که بتوانید به چیدمانِ شکل ۱ –ب برسید، به یکی از چیدمان‌های بهینه مسئله رسیده اید.



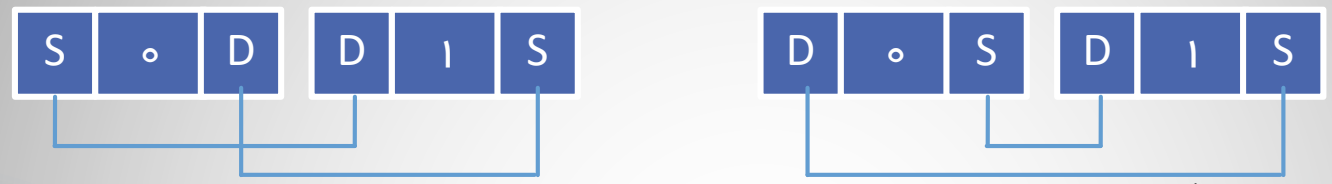
شکل ۱ –الف(۷تقاطع)



شکل ۱ –ب(۲ تقاطع)

نکات قابل توجه

- ترانزیستورهای یک ردیف می‌توانند به هم دیگر نیز اتصال داشته باشند، مانند شکل زیر.

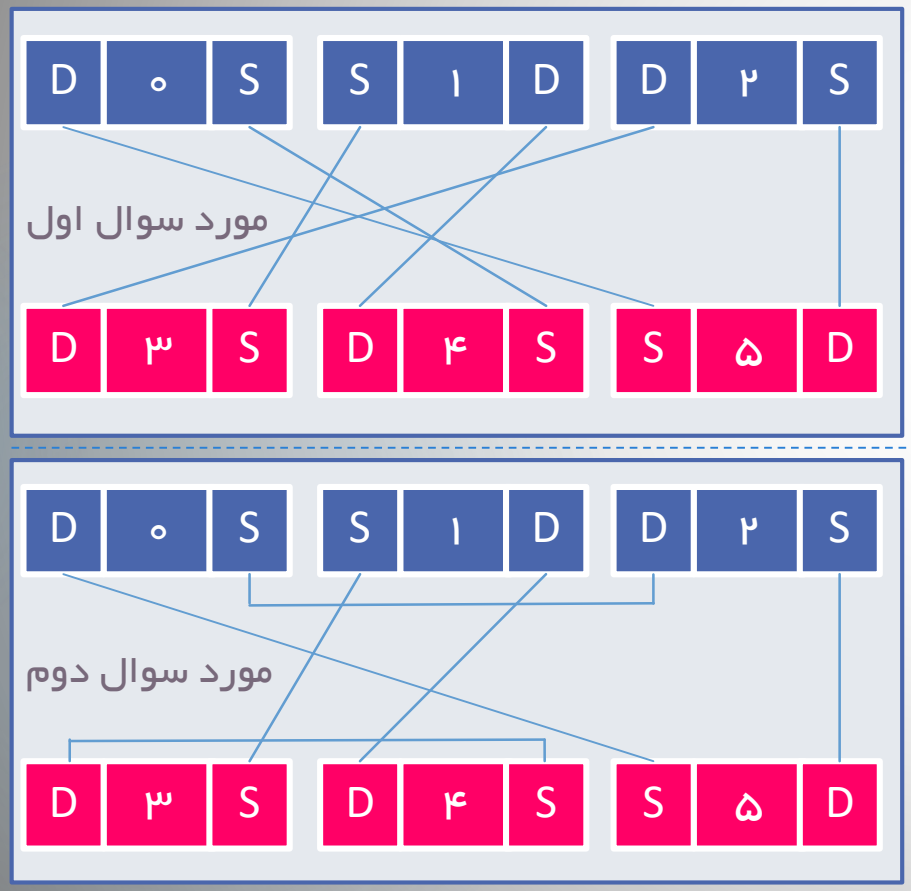


- با استفاده از الگوریتم‌های جستجوی محلی نظیر ژنتیک امکان دارد به بهترین جواب نرسید. توجه کنید که ما از شما جوابی خوب می‌خواهیم و لزومی ندارد به بهترین جواب برسید.
- شرط خاتمه الگوریتم ژنتیک فقط از بین بردن تمام تقاطع ها نیست و به ازای هر ورودی، مقداری بیشینه برای تعداد نسل‌های محاسبه‌شده داده می‌شود و در صورتی که بعد از رسیدن به آن مقدار هنوز تعدادی تقاطع باقی مانده باشد، بهترین حالت بدست آمده تا آن لحظه، خروجی برنامه است.
- دقت کنید که توابع Mutate و Crossover تغییرات زیر را روی چیپ ایجاد نکنند:
  - اضافه شدن تعداد ترانزیستورهای یک ردیف
  - ایجاد چند ترانریستور با یک نام
  - انتقال ترانزیستورها از ردیفی به ردیفی دیگر

ساختار ورودی(مثال در صفحه بعد)

- به ازای هر مورد سوال (test Case)
  - در خط اول بیشینه‌ی تعداد نسل‌ها برای محاسبه داده می‌شود.
  - در سطر بعد دو مقدار m و n به ترتیب برای تعدادِ ترانزیستورهای NMOS و PMOS داده می‌شود.
  - در m سطر بعد، در هر سطر دو جفت مقدار داده می‌شود، در i امین سطر از این m سطر:
    - جفت مقدار اول مشخص کننده شماره‌ی ترانزیستوری که پایه‌ی Source ترانزیستور i ام به آن متصل است و نوع پایه ای که به آن متصل است می‌باشد(۰برای Source و ۱ برای Drain).
    - جفت مقدار دوم مشخص کننده شماره‌ی ترانزیستوری که پایه‌ی Drain ترانزیستور i ام به آن متصل است و نوع پایه ای که به آن متصل است می‌باشد(۰برای Source و ۱ برای Drain).
  - در n سطر بعد، در هر سطر دو جفت مقدار داده می‌شود، در j امین سطر از این n سطر:
    - جفت مقدار اول مشخص کننده شماره‌ی ترانزیستوری که پایه‌ی Source ترانزیستور (j+m) ام به آن متصل است و نوع پایه ای که به آن متصل است می‌باشد(۰برای Source و ۱ برای Drain).
    - جفت مقدار دوم مشخص کننده شماره‌ی ترانزیستوری که پایه‌ی Drain ترانزیستور (j+m) ام به آن متصل است و نوع پایه ای که به آن متصل است می‌باشد(۰برای Source و ۱ برای Drain).

مثال ورودی




```
300
3 3
4 0 5 0
3 0 4 1
5 1 3 1
1 0 2 1
0 0 1 1
0 1 2 0
200
3 3
2 1 5 0
3 0 4 1
5 1 0 0
1 0 4 0
3 1 1 1
0 1 2 0
```

در "مورد سوال اول" پایه source ترانزیستور یکم به پایه source ترانزیستور سوم متصل است.

در "مورد سوال اول" پایه source ترانزیستور پنجم به پایه drain ترانزیستور صفرم متصل است.

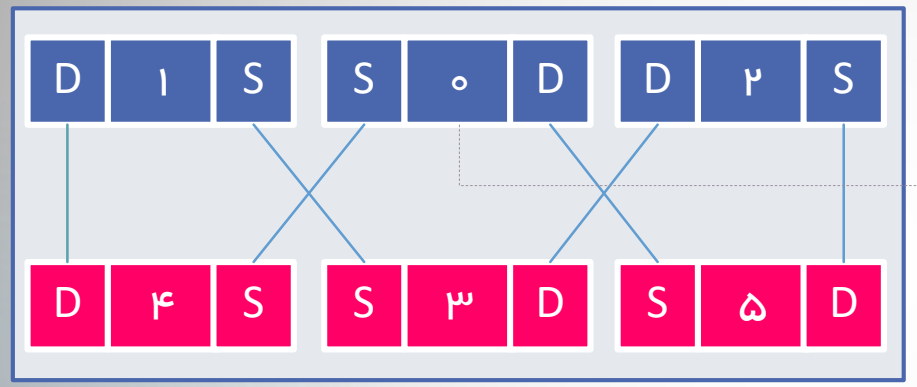
در "مورد سوال دوم" پایه drain ترانزیستور دوم به پایه source ترانزیستور صفرم متصل است.

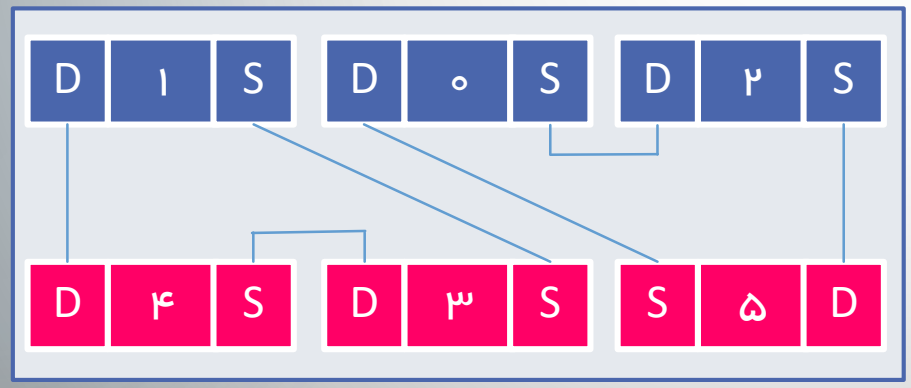
ساختار خروجی(مثال در صفحه بعد)

- به ازای هر مورد ورودی:
  - در سطر اول خروجی شماره‌ی نسلی که در آن به جواب رسیدید را چاپ کنید.
  - در سطر بعد تعداد تقاطع باقی‌مانده در بهترین حالت را چاپ کنید.
  - در سطر سوم ترتیب قرار گرفتن ترانزیستورها در لایه‌ی بالا (NMOS) و جهت قرار گرفتن هرکدام را چاپ کنید. (۰برای Source-Drain و ۱ برای Drain-Source)
  - در سطر چهارم ترتیب قرار گرفتن ترانزیستورها در لایه‌ی پایین (PMOS) و جهت قرار گرفتن هرکدام را چاپ کنید. (۰برای Source-Drain و ۱ برای Drain-Source)
- داده‌های یک سطر را با کاراکتر فاصله از هم جدا کنید.

مثال خروجی

- خروجی زیر به ازای مثال ورودی داده‌شده در همین سند است.






```
300
2
1 1 0 0 2 1
4 1 3 0 5 0
11
0
1 1 0 1 2 1
4 1 3 1 5 0
```

تحویل‌دادنی های پروژه

- مهلت مورد زیر ۶ آذر می‌باشد:
  - توضیح تئوری برنامه‌ای که برای پیاده‌سازی سوال دارید(توضیح نحوه‌ی پیاده سازی توابع Mutate Crossover و نحوه محاسبه fitness)
    - توضیح مختصر کافیست.(در حد یک‌رو از صفحه A4)
- مهلت موارد زیر ۱۲ آذر می‌باشد:
  - کدمنبع یا پروژه برنامه پیاده‌سازی شده.
  - یک فایل متنی (.txt) که خروجی برنامه به ازای مورد سوال‌های داده‌شده در آن باشد.(مورد سوال ها بعدا در سامانه درس افزار قرار داده خواهندشد.)
    - این فایل باید قبل از تحویل حضوری آماده‌شده باشد.(آماده‌شده آن یا به عبارتی اجرای برنامه به ازای مورد سوال‌ها ممکن است زمانبر باشد.)
    - فرمت فایل متنی باید حتما به صورت خواسته شده‌باشد.(این فایل با نرم‌افزار تست خواهد شد.)

موارد امتیازی

- موارد امتیازی اصلی:
  - *پیاده‌سازی امکان عوض کردن جهت ترانزیستور در ردیف خود.*
  - طراحی و پیاده‌سازی بهتر تابع‌های Mutate و Crossover
  - محاسبه مقداری برای مقایسه دو حالت، زمانی که تعداد تقاطع‌های برابر دارند.(محسابه بهترFitness)
  - یافتن سریعتر جواب بهینه.
  - یافتن جوابی بهتر در نسل داده‌شده.
- موارد امتیازی فرعی:
  - تهیه رابط گرافیکی مناسب برای نمایش جواب مسئله.

موفق باشید!