

به نام خدا

نام اعضای گروه:

هانیه رحیمی 40318103

حنانه وفایی نژاد 40330693

حدیث حیدرزاده 40316233

پروژه مدار منطقی :

ساده سازی توابع بولی با الگوریتم کوین مک-کلاسی (Quine-McCluskey)

پروژه چیست و ورودی / خروجی آن چیست؟

توضیحات:

در این پروژه، با استفاده از زبان Java و به کارگیری الگوریتم Quine-McCluskey، یک برنامه برای ساده سازی توابع بولی بر اساس مینترم های داده شده پیاده سازی شده است.

تعریف:

الگوریتم Quine-McCluskey یک روش دقیق و سیستماتیک برای ساده سازی توابع بولی است. برخلاف روش نقشه کارنو (Karnaugh Map) که محدود به متغیرهای کم است، این الگوریتم برای تعداد متغیرهای بالا نیز قابل استفاده است و در واقع نسخه عددی شده و قابل برنامه نویسی از آن محسوب می شود.

مزایا:

- دقت بالا و بدون خطا
- قابل پیاده سازی در زبان های برنامه نویسی
- مناسب برای توابع با تعداد متغیر بالا

معایب:

- پیچیدگی زمانی بالا برای توابع بزرگ
- نیاز به بهینه سازی بیشتر در مراحل پوشش

مراحل کلی:

1. تبدیل مینترم ها به نمایش دودویی
2. گروه بندی مینترم ها بر اساس تعداد 1 در نمایش دودویی
3. ترکیب گام به گام مینترم ها برای ساخت Prime Implicants (PI)
4. تشخیص Prime Implicants نهایی
5. در این پروژه صرفا تا مرحله تولید PI پیش میرویم

ورودی کاربر:

- تعداد متغیرها (بین 2 تا 6)
- لیستی از مینترم ها (عدد ده دهی بین 0 تا $2^n - 1$)

خروجی برنامه:

- نمایش عبارت ساده شده تابع بولی بر اساس مینترم ها، به صورت حروفی (مانند $A'B + AC$)

نحوه اجرای پروژه:

برای اجرای پروژه مراحل زیر را طی کنید:

- باز کردن فایل QM در محیط IDE
- اجرای برنامه
- وارد کردن تعداد متغیرها و مینترم ها طبق درخواست برنامه
- مشاهده خروجی ساده شده

نمونه اجرا:

QM ساده سازی مدار منطقی - روش

تعداد متغیرها (2 تا 6):

مینترمها (جداشده با کاما، مثل 0,1,2):

ساده سازی کن

عبارت ساده شده

$C' + B$

توضیح کامل درباره کدهای پروژه:

`getPrimeImplicants (List<String> minterms)`

- اجرای مرحله ترکیب مینترم ها برای یافتن Prime Implicant ها
- ترکیب رشته هایی که تنها در یک بیت تفاوت دارند
- بازگشت لیستی از ترکیب های نهایی که قابل ساده سازی نیستند
مثال: "000" و "001" → "00 -"

`getDifferencePosition(String a, String b)`

- بررسی اینکه دو رشته دودویی در " فقط یک بیت " تفاوت دارند یا نه
- اگر فقط یک تفاوت وجود داشته باشد، موقعیت آن بیت را (به صورت index عددی) بر می گرداند
- اگر بیش از یک تفاوت وجود داشته باشد، ترکیب امکان پذیر نیست (1- بر می گردد)

`binaryToLiteral (String bin, int numVars)`

- تبدیل رشته دودویی مانند " 0-10 " به فرم حروفی مانند " A'CD "
- از کاراکتر " - " برای متغیرهایی استفاده میشود که در ترکیب حذف شده اند
- حروف از A تا F به ترتیب برای نمایش متغیرها استفاده می شوند

ساختار کلی:

کد به صورت ساده و خوانا نوشته شده و از اصول clean code پیروی میکند.
در هر مرحله، وظایف به صورت جداگانه تعریف شده اند و با کامنت توضیح داده شده اند.