موضوع: تکنیک‌های واترمارکینگ در طراحی دیجیتال و مدارهای منطقی

مقدمه

در طراحی مدارهای دیجیتال و سامانه‌های سخت‌افزاری، حفاظت از مالکیت فکری (IP Protection) یکی از چالش‌های مهم به شمار می‌رود. یکی از روش‌های متداول برای اثبات مالکیت و جلوگیری از کپی‌برداری غیرمجاز، استفاده از واترمارکینگ سخت‌افزاری (Hardware Watermarking) است.

در این روش، طراح تلاش می‌کند بدون ایجاد تغییر محسوس در عملکرد اصلی مدار، یک امضای دیجیتال یا شناسه اختصاصی را در ساختار طراحی جاسازی کند؛ به‌گونه‌ای که در آینده بتوان از آن برای اثبات مالکیت، تشخیص جعل، یا اعتبارسنجی طراحی استفاده کرد.

در این تمرین با سه روش مختلف واترمارکینگ آشنا خواهید شد:

- استفاده از حالت‌های Don’t-Care در توابع منطقی
- جاسازی واترمارک در حالت‌های نامعتبر یک Encoder
- واترمارکینگ مبتنی بر گراف و درج یال(Edge Insertion)

هدف این تمرین آن است که دانشجویان علاوه بر درک مفاهیم تئوری، بتوانند نحوه جاسازی، بازیابی و تحلیل تاثیر واترمارک بر عملکرد مدار را نیز بررسی کنند.

نکات مهم

- تمامی مراحل حل به‌صورت کامل و مرحله به مرحله نوشته شود.
- در صورت استفاده از جدول کارنو، نحوه گروه‌بندی‌ها مشخص گردد.
- توضیحات تحلیلی بخش مهمی از نمره را تشکیل می‌دهند.
- صرفاً ارائه پاسخ نهایی بدون روند حل، امتیاز کامل نخواهد داشت.
- در تمامی سوالات فرض کنید هدف اصلی، جاسازی واترمارک بدون تغییر عملکرد عادی مدار است.

سوال ۱- واترمارکینگ با استفاده از شرایط Don’t-Care (۴۵ نمره)

در بسیاری از طراحی‌های منطقی، برخی حالات ورودی هیچ‌گاه در عملکرد عادی مدار استفاده نمی‌شوند و یا مقدار خروجی آن‌ها اهمیتی ندارد. این حالت‌ها با عنوان Don’t-Care شناخته می‌شوند و می‌توان از آن‌ها برای جاسازی اطلاعات اضافی بدون تغییر عملکرد اصلی مدار استفاده کرد.

در این سوال هدف آن است که با استفاده از حالات Don't-Care موجود در تابع منطقی، یک امضای دیجیتال در مدار جاسازی شود. سپس تاثیر این فرآیند بر فرم نهایی تابع و نحوه بازیابی واترمارک بررسی خواهد شد.

تابع منطقی زیر را در نظر بگیرید:

$$F(A,B,C,D,E) = \sum m(0,3,7,10,12,15,27) + d(5,8,19,22,23)$$

فرض کنید قصد داریم امضای ۵ بیتی «۰۱۱۰۰» را با استفاده از روش واترمارکینگ مبتنی بر شرایط Don't-Care در تابع فوق جاسازی کنیم.

خواسته‌ها

۱. مقادیر متناظر با مینترم‌های Don't-Care را به‌گونه‌ای تغییر دهید که امضای موردنظر در تابع جاسازی شود.
۲. مینترم‌های ON تابع واترمارک‌شده F_{wm} را مشخص کنید.
۳. تابع F_{wm} را با استفاده از نقشه کارنو مینیمم کرده و فرم ساده‌شده SOP را به‌دست آورید.
۴. به‌صورت مختصر توضیح دهید:
   - چرا عملکرد عادی مدار دچار اختلال نمی‌شود؟
   - چگونه می‌توان واترمارک جاسازی‌شده را در آینده استخراج و صحت‌سنجی کرد؟

بخش عملی)

در این بخش، شما باید با استفاده از حالات Don't-Care موجود در تابع منطقی مطرح شده در بخش اول، یک واترمارک دیجیتال در سطح RTL جاسازی کنید و سپس با استفاده از شبیه‌سازی و تست‌بنچ، وجود آن را مورد بررسی و اعتبارسنجی قرار دهید.

در ابتدا، یک ماژول Verilog مربوط به تابع منطقی ذکر شده در اختیار شما قرار گرفته است:

```
module watermark_example (
    input  wire A,
    input  wire B,
    input  wire C,
    input  wire D,
    input  wire E,
    output wire F
);

assign F =
       (~A & ~B & ~C & ~D & ~E) |    // m0
```

```
        (~A & ~B & ~C &  D &  E) |    // m3
        (~A & ~B &  C &  D &  E) |    // m7
        (~A &  B & ~C &  D & ~E) |    // m10
        (~A &  B &  C & ~D & ~E) |    // m12
        (~A &  B &  C &  D &  E) |    // m15
        ( A &  B & ~C &  D &  E);     // m27

endmodule
```

هدف شما این است که با استفاده از حالات Don’t-Care ، واترمارک دیجیتال «۰۱۱۰۰» را در کد ماژول وریلاگ داده شده اضافه کنید و سپس وجود آن را همان طور که پیش تر هم اشاره شد با استفاده از شبیه‌سازی و تست‌بنچ اثبات کنید.

خواسته ها:

۱. با استفاده از حالات Don’t-Care ، واترمارک ۵ بیتی زیر را در کد وریلاگ داده شده جاسازی کنید:

01100

۲. نسخه واترمارک‌شده مدار را به زبان Verilog پیاده‌سازی کنید .

۳. یک Testbench کامل برای مدار طراحی کنید به‌گونه‌ای که:

- عملکرد صحیح مدار را بررسی کند .
- وجود واترمارک را Verify کند .
- نشان دهد عملکرد اصلی مدار دچار تغییر نشده است .

راهنمایی

- تنها مجاز به تغییر حالات Don’t-Care هستید .
- برای Verify کردن واترمارک، کافی است خروجی مدار را در حالات Don’t-Care بررسی کنید .
- پیشنهاد می‌شود در تست‌بنچ تمامی حالات ورودی تست شوند .
- می‌توانید از دستور $display برای نمایش و بازیابی واترمارک استفاده کنید .
- هدف اصلی این تمرین، شبیه‌سازی فرآیند واقعی اثبات مالکیت IP در صنعت نیمه‌هادی است.

بخش اختیاری(Bonus)

فرض کنید این IP متعلق به یک شرکت طراحی تراشه بوده و بعدها در یک محصول تجاری مشکوک به کپی‌برداری مشاهده شده است.

به سوالات زیر پاسخ دهید:

۱. چگونه واترمارک می‌تواند در فرآیند اثبات مالکیت حقوقی مورد استفاده قرار گیرد؟
۲. مهاجم چگونه ممکن است تلاش کند واترمارک را حذف یا مخفی کند؟
۳. چه روش‌هایی می‌توانند مقاومت واترمارک را در برابر حذف یا تغییر افزایش دهند؟

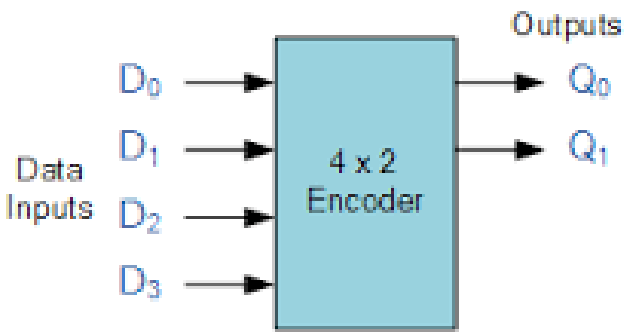
سوال ۲- واترمارکینگ در Encoder چهار به دو (۳۵ نمره)

در مدارهای ترکیبی مانندEncoder ها معمولاً تنها بخشی از حالات ورودی معتبر هستند و سایر ترکیب‌ها در عملکرد واقعی مدار مورد استفاده قرار نمی‌گیرند. این ویژگی امکان استفاده از حالت‌های نامعتبر را برای جاسازی واترمارک فراهم می‌کند.

در این سوال قرار است با استفاده از حالت‌های Don't-Care موجود در یک Encoder چهار به دو، یک رشته متنی به‌صورت بیت‌های ASCII در مدار جاسازی شود. هدف اصلی این تمرین، آشنایی با روش نگاشت اطلاعات باینری روی حالت‌های نامعتبر مدار و تحلیل تاثیر آن بر ساختار منطقی مدار است.

در یک Encoder چهار به دو، تنها حالتی معتبر است که دقیقاً یکی از ورودی‌های a, b, c, d برابر ۱ باشد؛ بنابراین سایر ۱۲ حالت ورودی به‌عنوان Don't-Care در نظر گرفته می‌شوند.

رشته سه‌حرفی «HST» را با استفاده از روش تدریس‌شده در کلاس در الگوهای Don't-Care جاسازی کنید.



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

خواسته‌ها

۱. رشته ASCII بیست‌وچهاربیتی متناظر با «HST» را بنویسید.
۲. از سمت راست، دنباله بیتی را به گروه‌های ۵ بیتی تقسیم کنید.
   - در صورتی که طول گروه آخر کامل نبود، صفرهای لازم را به انتهای آن اضافه کنید.
۳. زوج‌های جدید ورودی–خروجی را تعیین کرده و جدول درستی به‌روزشده را ارائه دهید.

۴. با استفاده از جدول کارنو برای خروجی‌های XX و YY، حداقل ترم‌های اضافی موردنیاز را به‌دست آورده و توضیح دهید چرا عملکرد اصلی Encoder بدون تغییر باقی می‌ماند.

سوال ۳- واترمارکینگ در مسئله رنگ‌آمیزی گراف (۳۰ نمره)

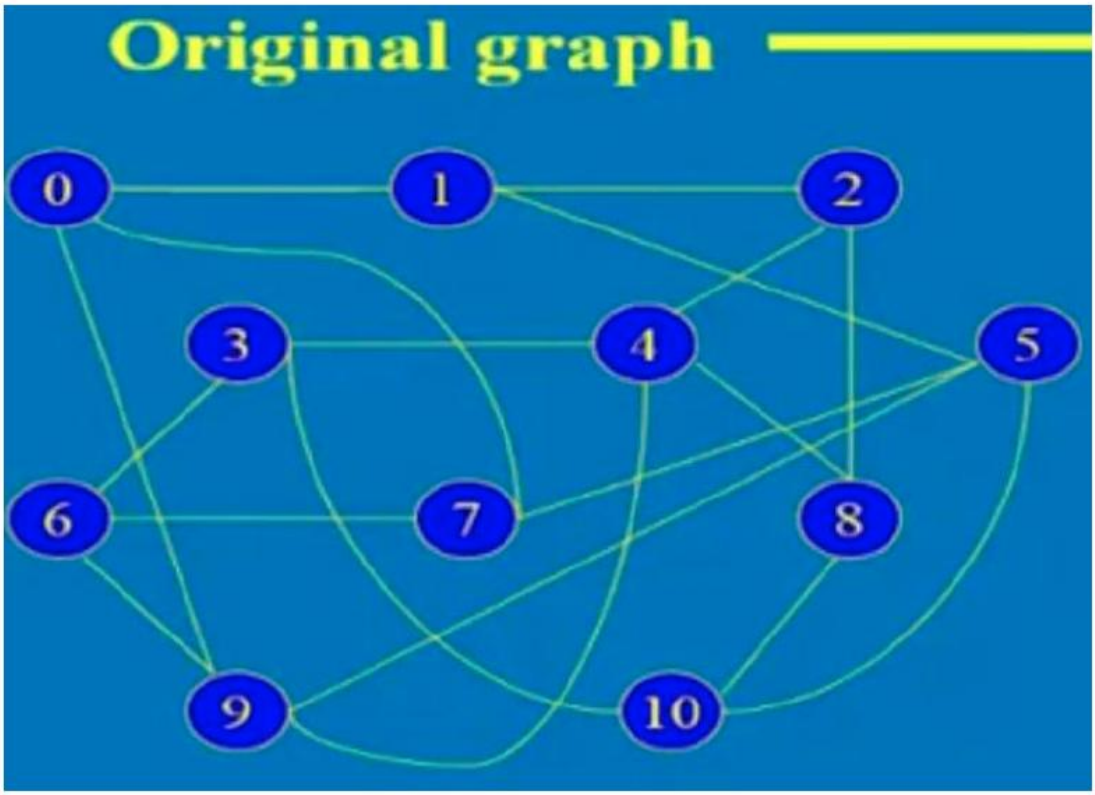
یکی از روش‌های مطرح در واترمارکینگ ساختارهای محاسباتی، استفاده از ویژگی‌های گراف و تغییر کنترل‌شده در ارتباط بین راس‌ها است. در این روش، اطلاعات واترمارک از طریق اضافه‌کردن یال‌ها طبق یک قاعده مشخص در ساختار گراف جاسازی می‌شود.

در این سوال هدف آن است که با استفاده از روش Edge Insertion ارائه‌شده در کلاس، یک رشته باینری در گراف داده‌شده جاسازی گردد و سپس امکان استخراج و اعتبارسنجی آن بررسی شود.

با استفاده از روش درج یال (Edge Insertion Rule) که در کلاس ارائه شد، رشته باینری زیر را در گراف داده‌شده جاسازی کنید:

1101010010110011

سپس مراحل استخراج و اعتبارسنجی واترمارک را توضیح دهید.



نکات کلی گزارش

• اگر در انجام هر یک از قسمتهای تمرین فرض اولیهای را در نظر گرفتهاید، حتماً مفروضات خود را در گزارش کار ذکر کنید دقت کنید که مفروضات شما نباید به گونه ای باشند که ساختار کلی سوال را عوض کنند.

• نحوه ارزیابی گزارش و نمره دهی در انتهای این فایل ارائه شده است.

بخش قابل توجهی از نمره‌ی این تمرین به گزارش کار آن اختصاص دارد ؛ لذا برای نوشتن آن زمان کافی اختصاص دهید.

• صفحه‌ی اول گزارش کار باید سربرگ رسمی دانشکده باشد. گزارش کار باید شامل فهرست و شماره صفحه بوده و تمامی قسمتهای آن به صورت تایپ شده باشد. تمامی تصاویر باید دارای زیرنویس و تمامی جداول دارای بالانویس باشند .

• تمامی نمودارها، جداول و نتایج بایستی توضیح داده و تحلیل شوند.

• فایلهای شبیه‌سازی را در یک پوشه مجزا قرار داده و گزارش را به دو فرمت WORD و PDF در کنار این فایلها فشرده کرده و با نام CA3_HWS_StudentID ارسال کنید.

• در صورت داشتن هرگونه سوال با ایمیل Behzadjannati@ut.ac.ir یا PV پیام رسان بله در ارتباط باشید یا سوال خود را در گروه درسی در نرم افزار بله مطرح کنید.

موفق باشید

بخش	خواسته	نمره
سوال اول	نوشتن تابع F بدون واترمارک	۲ نمره
	جایگذاری واترمارک در استیت های Don't Care	۵ نمره
	ساده سازی تابع به همراه واترمارک با جدول کارنو	۱۵ نمره
	جایگذاری واترمارک در ماژول وریلاگ داده شده و نوشتن تست بنچ	۱۵
	پاسخ به سوالات بخش امتیازی	۸ نمره
بخش دوم	ایجاد بیت استریم بر اساس حروف اسکی داده شده	۳ نمره
	تقسیم به دسته های ۵ تایی	۵ نمره
	پیاده سازی واترمارک در استیت های Don't Care جدول درستی	۱۴ نمره
	ساده سازی با جدول کارنو و نوشتن تابع با واترمارک	۱۰ نمره
	توضیح چگونگی کارکرد واترمارک و عدم تغییر فانکشن اصلی مدار	۳ نمره
بخش سوم	توضیح روش حل	۵
	توضیح نحوه اضافه شدن یالها در هر مرحله	۱۵
	گراف رسم شده نهایی به همراه یالهای جدید	۱۰
		مجموع ۱۱۰