طراحی و بهینه سازی مدارات منطقی ترتیبی سنکرون با استفاده از سخت افزارهای تکامل پذیر

پریسا سلیمانی[١] ، کریم محمدی[٢] ، رضا صباغی ندوشن[٣] ، ستار میرزاکوچکی[٤]

[١] کارشناسی ارشد، دانشگاه آزاد اسلامی واحد تهران مرکزی، گروه مهندسی برق، تهران، ایران. Email: parisa.soleimani@gmail.com

[٢] استاد، دانشگاه علم و صنعت ایران، گروه مهندسی برق، تهران، ایران

[٣] استادیار، دانشگاه آزاد اسلامی واحد تهران مرکزی، گروه مهندسی برق، تهران، ایران

[٤] دانشیار، دانشگاه علم و صنعت ایران، گروه مهندسی برق، تهران، ایران

چکیده

یکی از شاخه های سیستم های الهام گرفته از طبیعت، سخت افزارهای تکامل پذیر[١](EHW) است. سخت افزارهای تکامل پذیر روشی نوین برای طراحی مدارات منطقی دیجیتال می باشند. در این مقاله، روشی به منظور طراحی مدارات منطقی ترتیبی سنکرون با استفاده از سخت افزارهای تکامل پذیر ارائه شده است. در این رویکرد، مدار منطقی ترتیبی مورد نظر، به دو بخش تقسیم می شود: بخش ترکیبی مدار و D فلیپ فلاپ ها. بخش منطقی ترکیبی، توسط ساختاری ثابت طراحی شده و اتصالات این بخش توسط الگوریتم ژنتیک تنظیم می شوند. نتایج نشان می دهند که روش ما قادر است تا تعداد گیت های تشکیل دهنده مدار و نیز متوسط تعداد نسل های اجرای الگوریتم را کاهش دهد.

کلید واژه: مدارات ترتیبی، مدارات ترکیبی، الگوریتم ژنتیک، سخت افزارهای تکامل پذیر، الگوریتم های تکاملی.

© 2012 IAUCTB-IJSEE Science. All rights reserved

[1] Evolvable Hardware

Archive of SID

www.SID.ir

۱. مقدمه

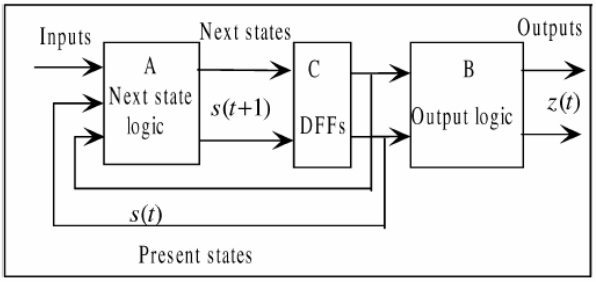
سخت‌افزار تکامل‌پذیر، سخت‌افزاری است که می تواند ساختار و رفتارش را به طور پویا و خودکار[۱۲] با تعامل با محیط اطرافش تغییر دهد [۱و۲]. این سخت افزار، الگوریتم های تکاملی را به منظور طراحی و بهینه سازی مدارات الکترونیکی بکار می گیرد. یکی از پرکاربرد ترین الگوریتم تکاملی که در این مقاله نیز از آن استفاده نموده ایم، الگوریتم ژنتیک است. این الگوریتم با انجام عملیات جستجو در میان اعضای مختلف، بر اساس یک تابع که بیانگر کارایی هر عضو می باشد، سعی در پیدا کردن بهترین راه حل برای طراحی مدار مورد نظر دارد. این الگوریتم برای رسیدن به این اهداف از عملیات انتخاب، تقاطع و جهش استفاده می نماید.

تاکنون در سیستم های دیجیتالی، تکامل مدارات ترتیبی نسبت به مدارات ترکیبی رشد کمتری داشته است و این نیز به دلیل ساده تر بودن ساختار مدارات ترکیبی نسبت به مدارات ترتیبی، و فاقد فیدبک بودن این مدارات می باشد. با این وجود، تلاش های کمی نیز بر روی مدارات ترتیبی صورت گرفته است. برای مثال، هیگوچی[۱۳] از الگوریتم ژنتیک برای جستجوی مدارهایی که تابع انتقال حالت مطلوب را نمایش می دهند، استفاده نمود[۳] ، مانوویت[۱٤]، مدارهای آشکار ساز فرکانس و شمارنده ۵ بیتی و نیز جمع کننده سریال را سنتز کرد[٤]. اپورتیوان[۱۵]، مدارهای جمع کننده سریال و آشکار ساز دنباله ۰۱۰۱ را با استفاده از الگوریتم ژنتیک تکامل داد[۵]. سولیمون[۱٦]، مدار شمارنده ۳ بیتی را طراحی نمود [٦] و شنتیا[۱۷] نیز مدارهای lion و شمارنده ٦ بیتی را تکامل داد[۷].

این مقاله، روشی را به منظور طراحی و بهینه سازی مدارات منطقی ترتیبی سنکرون با عملکرد صحیح ۱۰۰٪ و کمترین گیت های منطقی بکار رفته در آنها پیشنهاد می کند. در ادامه این مقاله در : بخش دوم ایده اصلی روش پیشنهادی توضیح داده می شود، بخش سوم عملگرهای الگوریتم ژنتیک را توصیف می نماید، بخش چهارم جزئیات فرآیند تعیین ساختار کروموزوم را تشریح می کند، بخش پنجم فرآیند ارزیابی مقدار برازش[۱۸] را برای مدارات تکاملی توصیف می نماید. بخش ششم بستر شبیه سازی استفاده شده به منظورتکامل مدارات، معرفی می شود. در بخش هفتم نتایج شبیه سازی را برای چند مدار آزمایش شده نمایش داده و در نهایت نتیجه گیری کلی دربخش هفتم آورده شده است.

۲. روش پیشنهادی

شکل۱، اجزاء تشکیل دهنده مدارات منطقی ترتیبی که از دو بخش مدار منطقی ترکیبی و D فلیپ فلاپ ها تشکیل شده اند را نشان می دهد. در این رویکرد، به منظور طراحی بخش های ترکیبی، یک نوع آرایه مستطیلی متشکل از گیتهای منطقی ارائه می کنیم و از آن برای ساختن حالت بعدی D فلیپ فلاپ ها و خروجی های اصلی مدار استفاده می نماییم.



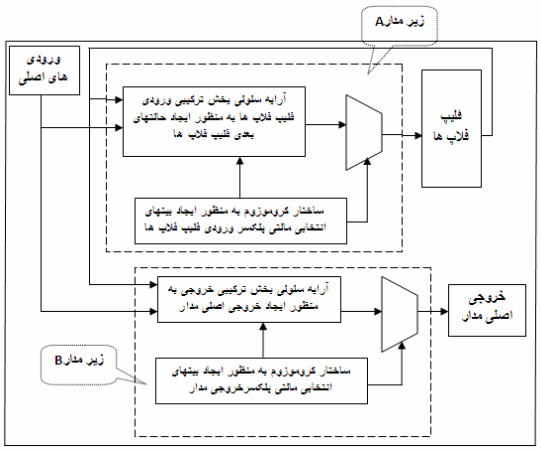
شکل۱: توصیف اجزاء تشکیل دهنده مدارات ترتیبی سنکرون[۸]

شکل۲، بلوک دیاگرام روش پیشنهادی را نشان می دهد.

[2] Autonomously
[3] Higuchi
[4] Manovit
[5] Aporntewan
[6] Solimon
[7] Shanthia
[8] Fitness

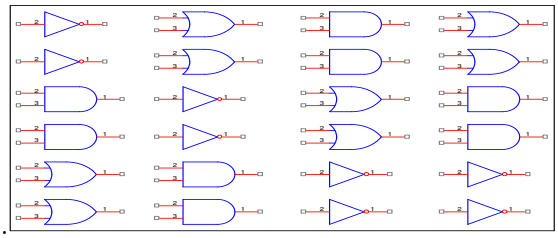
Archive of SID

www.SID.ir



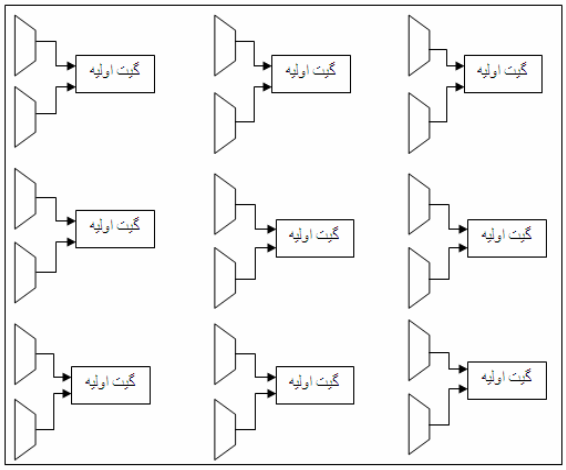
شکل۲: بلوک دیاگرام روش پیشنهادی

در شکل ۳ نیز آرایه سلولی پیشنهادی که دارای R سطر و C ستون می باشد و گیتهای منطقی تشکیل دهنده آن گیت های OR، AND و NOT می باشند، نمایش داده شده است. بجز گیت NOT، سایر گیت ها، دارای دو ورودی و یک خروجی هستند و ورودی هرگیت از ورودی های اصلی مدار، خروجی های D فلیپ فلاپ ها و یا خروجی گیت های قرار گرفته در ستون مجاور سمت چپ آن گیت، گرفته می شود.



شکل۳: شماتیک ساختار آرایه مستطیلی استفاده شده به منظور ساخت بخش های ترکیبی مدار ترتیبی

در این روش، به ورودی هر یک از گیت های منطقی موجود در هر آرایه سلولی، ورودی D فلیپ فلاپ ها و قبل از خروجی های اصلی مدار، یک مالتی پلکسر اضافه می شود(همانند شکل ٤). سپس با تعیین ساختار پیشنهادی رمزگذاری[19] کروموزوم(بخش ٤) و با استفاده از الگوریتم ژنتیک، حالتهای مختلف اتصال گیت ها برای هر آرایه بطور جداگانه ارزیابی می شوند تا به عملکرد صحیح مدار با حداقل تعداد گیت های منطقی بکار رفته شده در آن دست یابیم و در نهایت، مدار نهایی اسمبل می شود. ساختار مالتی پلکسر استفاده شده، در شکل ۵ نمایش داده شده است. ارزیابی هر آرایه سلولی بصورت مجزا، منجر به افزایش سرعت تکامل مدار و کاهش زمان اجرای برنامه می شود.



شکل۴: بلوک دیاگرام آرایه سلولی پس از اضافه نمودن مالتی پلکسر به آرایه

۳. پارامترهای الگوریت ژنتیک

در این مقاله، الگوریتم ژنتیک به عنوان الگوریتم تکاملی استفاده شده به منظور تکامل مدارات منطقی ترتیبی دیجیتال بکار رفته است. اعضای جمعیت از نوع رشته های بیتی بوده و پارامترهای الگوریتم به صورت زیر انتخاب شده اند:

انتخاب: در این تکامل، متد انتخابی به نام "انتخاب چرخ-رولت غیر تصادفی استاندارد" برای انتخاب والدین بکار گرفته شده است.

تقاطع: متد تقاطعی که در این کار مورد استفاده قرار گرفته است، تقاطع پراکنده نام دارد.

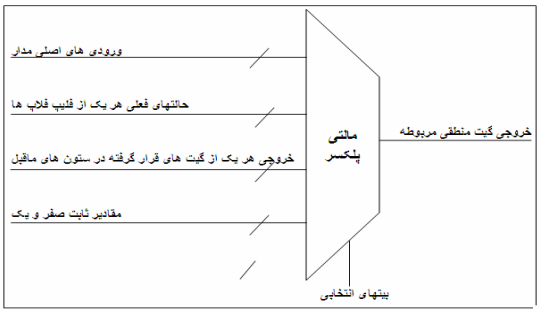
جهش: جهش دگرگونی اتفاقی ژن های درون کروموزوم را توصیف می نماید. متد جهشی که در اینجا بکار گرفته شده است ، جهش یکنواخت می باشد.

همچنین سایز جمعیت ۱۰ و ماکزیمم تعداد نسل های اجرای الگوریتم، ۳۰۰۰۰ نسل در نظر گرفته شده است. و در صورتی که پس از ۱۰۰۰۰ بار اجرای متوالی الگوریتم، بهبودی در مقدار برازندگی حاصل نشود، الگوریتم متوقف خواهد شد.

[9]Coding

Archive of SID

www.SID.ir



شکل ٥: ساختار مالتی پلکسر

٤. رمزگذاری کروموزوم

کروموزوم، ساختار مدار و اتصالات بین گیت های منطقی را تعیین می کند. در این رویکرد، در ورودی هر یک از گیت ها و D فلیپ فلاپ ها و قبل از خروجی های اصلی مدار، یک مالتی پلکسر قرار می دهیم. با تغییر بیت های انتخابی مالتی پلکسرها، اتصال بین گیت ها و D فلیپ فلاپ ها را تغییر می دهیم. ورودی مالتی پلکسرهای قرار گرفته در پایه های ورودی گیت های منطقی از ورودی های اصلی مدار، حالت های فعلی فلیپ فلاپ ها، خروجی تمامی گیت های منطقی قرار گرفته در ستون مجاور سمت چپ گیت مورد نظر و نیز مقادیر ثابت ۰ و ۱ گرفته می شود. همچنین ورودی مالتی پلکسرهای قرار گرفته در ورودی D فلیپ فلاپ ها و نیز قبل از خروجی های اصلی مدار، از ورودی های اصلی مدار و خروجی تمامی گیت های منطقی قرار گرفته در ستون های ما قبل از خود گرفته می شوند. تغییر بیت های انتخابی مالتی پلکسرها، منجر به اتصالات مختلف ما بین گیت های منطقی آرایه سلولی شده و مدارات متفاوتی را ایجاد می کند. بنابراین در این رویکرد ما از بیت های انتخابی مالتی پلکسرها به عنوان ژن های کروموزوم استفاده کرده ایم(شکل ٦).

٥. فرآیند ارزیابی مقدار برازندگی

در این بخش فرآیند ارزیابی مقدار برازندگی توضیح داده شده است. در اینجا به منظور ارزیابی مدارات ترتیبی، از ماشین حالت استفاده کرده ایم. در این روش، ابتدا حالت مورد نظر به ورودی فلیپ فلاپ ها تخصیص داده می شود، سپس با تغییر ورودی های اصلی مدار، خروجی مدار اندازه گیری و با خروجی مطلوب مقایسه می شود. در هر حالت خروجی مطلوب برای مدار ترکیبی بخش ورودی، حالت بعدی فلیپ فلاپ ها است. در حالی که خروجی مطلوب برای مدار ترکیبی بخش خروجی، خروجی اصلی مدار می باشد. اگر در هر حالت خروجی بدست آمده با خروجی مطلوب برابر باشد، مقدار برازندگی افزایش می یابد.

در روش پیشنهادی، ارزیابی مدار مورد نظر در دو مرحله انجام می شود:

١.مرحله طراحی ٢. مرحله بهینه سازی

در مرحله اول، صحت عملکرد مدار ارزیابی می شود. هدف از این مرحله، رسیدن به صحت عملکرد ١٠٠٪ است. سپس در مرحله دوم، بهینه سازی توسط کاهش تعداد گیتهای منطقی بکار رفته در مدار انجام می شود. این مرحله زمانی قابل انجام است که در مرحله اول به صحت عملکرد ١٠٠٪ رسیده باشیم.

در مرحله طراحی، هر عضو جمعیت انتخابی توسط مراحل ذیل ارزیابی می شود:

١. مقدار اولیه تابع برازندگی برابر با صفر در نظر گرفته می شود.
٢. مقادیر مورد نظر، به ورودی های اصلی مدار و ورودی فلیپ فلاپ ها تخصیص داده شده و سپس با ارسال سیگنال کلاک به مدار، خروجی مدار اندازه گیری می شود.
٣. خروجی بدست آمده از مدار با خروجی مطلوب مقایسه شده و در نهایت مقدار برازندگی از رابطه ذیل ارزیابی می شود.

تعداد بیتهای مساوی با خروجی مطلوب+$F_{اولیه}$ =$F_{طراحی}$

٤. مراحل ٢ و ٣ برای باقیمانده حالت های ماشین حالت مورد نظر تکرار شده و عملکرد مدار ارزیابی می شود.

مقدار برازندگی برای مرحله بهینه سازی به صورت ذیل ارزیابی می شود:

١. مقداراولیه تابع برازندگی در این حالت برابر می شود با :

Archive of SID

www.SID.ir

$F_{بهینه سازی}=R*C$

۲. برای هر یک از اعضاء جمعیت، تعداد گیت های استفاده شده در مدار اندازه گیری می شود. بدین معنی که :

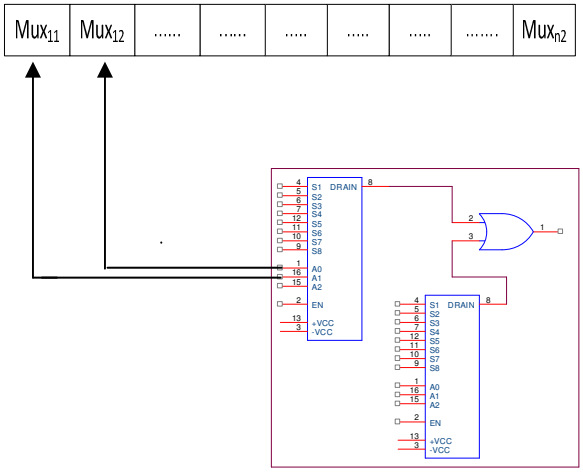
$F_{بهینه سازی}= (R*C)$ - تعداد گیت های استفاده شده در مدار جدید ، C، تعداد سطر های آرایه سلولی و Rکه در رابطه بالا تعداد ستون های آرایه سلولی می باشد.

در نهایت، مقدار نهایی برازندگی برای مدار مورد نظربرابر است با:

$F_{کل}= F_{طراحی} + F_{بهینه سازی}$

هر دو فرآیند توصیف شده در بالا، برای ارزیابی هر دو بخش ترکیبی مدار ترتیبی بکار گرفته می شود.

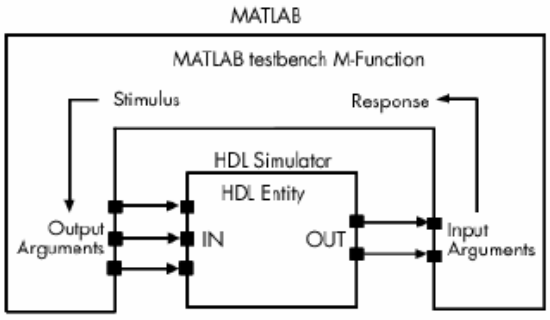
اصلاح بینایی توسط اکسایمر لیزر روزبروز گسترش یافته و هدف آن از بین بردن خطاهای انکساری و بخصوص نزدیک بینی با دو تکنیک رایجPRK و LASEK می باشد در این پروژه هدف مدل کردن فرآیند رفع عیوب انکساری چشم توسط اکسایمر لیزر بود تا بتوان به قابلیت پیش بینی مناسبی در قبل از انجام عمل جراحی رسید و از آن در جهت انتخاب پارامترها و دقت بیشتر و ارزیابی عمل قبل ازجراحی استفاده نمود که به این هدف رسیده شد.



شکل ٦: ساختار پیشنهادی رمزگذاری کروموزوم

٦. بستر شبیه سازی

در این مقاله، از Modelsim به عنوان شبیه ساز زبان برنامه نویسی سخت افزار VHDL، و برای تکامل مدار از جعبه ابزار[۲۰] الگوریتم ژنتیک موجود در نرم افزار متلب استفاده شده است. همچنین از جعبه‌ابزار دیگری در متلب، با عنوان Simulator link ^{TM} MQ که امکان دسترسی به Modelsim از جمله باز کردن کد HDL، اجرای آن به ازای ورودی‌های مختلف تعیین شده در متلب و ذخیره خروجی‌های آن در متغیرهای این نرم افزار را می‌دهد، نیز استفاده شده است. شکل۷ بلوک دیاگرام این فرآیند را نشان می دهد.



شکل۷: ساختار کلی ارتباط متلب با Modelsim [۹]

بنابراین این جعبه ابزار، به عنوان لینک ارتباطی ما بین متلب و Modelsim عمل می نماید.

۷. آزمایشات و نتایج

در این بخش، روش پیشنهادی بر روی دو نوع مدار ترتیبی مختلف آزمایش و نتایج بدست آمده از آنها، شرح داده شده است و در نهایت این نتایج با نتایج حاصل از کارهای مشابه قبلی مقایسه شده است.

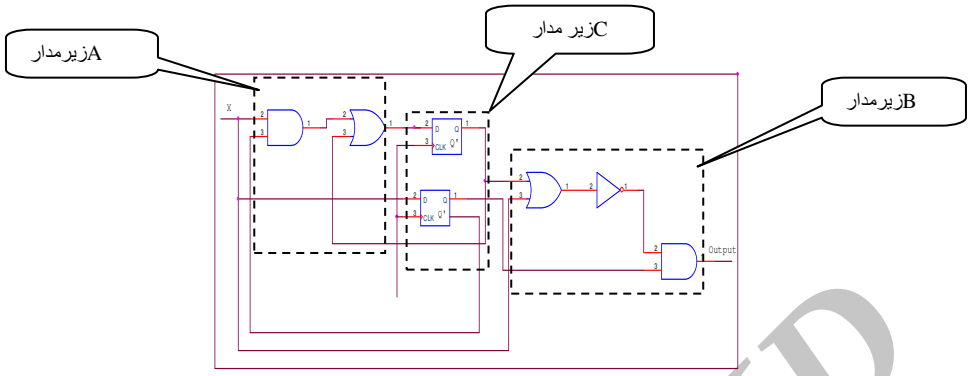
الف) مدار آشکار ساز دنباله ۱۰۱۰

در این بخش، روش پیشنهادی برروی یک مدار آشکار ساز دنباله ۱۰۱۰ آزمایش شده است. گراف انتقال حالت این مدار در شکل ۸ نمایش داده شده است.

[10] Toolbox

Archive of SID

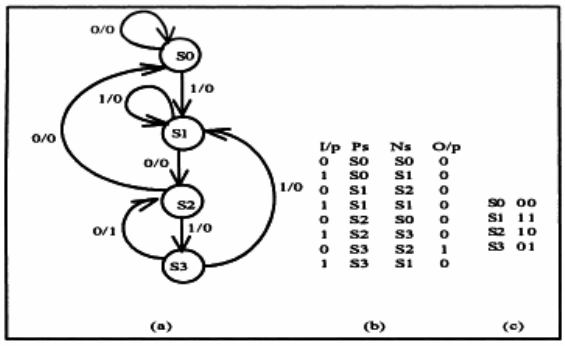
www.SID.ir



شکل۹: مدار تکامل یافته بدست آمده توسط روش پیشنهادی برای آشکار ساز دنباله ۱۰۱۰

جدول۱: جدول حالت مدار آشکار ساز دنباله ۱۰۱۰، نمایش حالت‌های آن به صورت کد باینری و تجزیه آن به دو زیر مدار A و B به منظور طراحی بخش‌های ترکیبی ورودی و خروجی

i/p	Ps	Ns	o/p		STT of the circuit	STT of subcircuit A	STT of subcircuit B
0	S0	S0	0		.i 3	.i 3	.i 3
1	S0	S1	0		.o 3	.o 2	.o 1
0	S1	S2	0		.p 8	.p 8	.p 8
1	S1	S1	0	S0=00	000 000	000 00	000 0
0	S2	S0	0	S1=11	100 110	100 11	100 0
1	S2	S3	0	S2=10	011 100	011 10	011 0
0	S3	S2	1	S3=01	111 110	111 11	111 0
1	S3	S1	0		010 000	010 00	010 0
					110 010	110 01	110 0
i/p	inputs				001 101	001 10	001 1
o/p	outputs				101 110	101 11	101 0
Ps	Present state						
Ns	Next state			Step1	Step2	Step3	



شکل۸: دیاگرام حالت مدار آشکار ساز دنباله ۱۰۱۰[۸]

در مرحله نخست جدول۱، ماشین باحالات محدود[۲۱] این مدار و مقادیر تخصیص داده شده به هر حالت نمایش داده شده است. در مرحله دوم این جدول، جدول انتقال حالت مدار مورد نظر نشان داده شده است. و در نهایت، در مرحله سوم ، این جدول انتقال حالت به دو بخش زیر مدار ترکیبی ورودی A و زیر مدار ترکیبی خروجی B تقسیم و نمایش داده شده است(همانند مقاله [۸]). از آنجا که این مدار، دارای چهار حالت است، به منظور ساختن آن نیاز به استفاده از دو فلیپ فلاپ می باشد. همانطور که در بخش های قبلی اشاره شد، در این روش هر یک از دو زیرمدار A و B بصورت جداگانه ارزیابی شده و در نهایت اسمبل شده و مدار نهایی بدست می آید.

همانطور که مشاهده می نمایید مدار نهایی که در شکل۹ نمایش داده شده است دارای دو گیت در زیر مدار ورودی A و سه گیت در زیر مدار خروجی B می باشد.

نتایج بدست آمده توسط روش پیشنهادی در مقایسه با[۱۰] در جدول ۲ نمایش داده شده است. در این آزمایش، ماکزیمم تعداد نسل ها به منظور تکامل زیر مدار ورودی A، ۱۰۲۰۰ نسل و به منظور تکامل زیر مدار خروجی B، ۷۱۵۰ نسل می باشد. این نتایج پس از ۲۰ بار اجرای الگوریتم بدست آمده اند که در مقایسه با روش پیشنهادی در[۱۰]، روش ما توانسته است در مدت زمان کمتر و با تعداد نسل های کمتری به صحت عملکرد ۱۰۰٪ رسیده و نیز زمان بهینه سازی مدار و تعداد گیت های سازنده آن، به نحو قابل ملاحظه ای کاهش یابد.

[11] Finite State Machine

Archive of SID

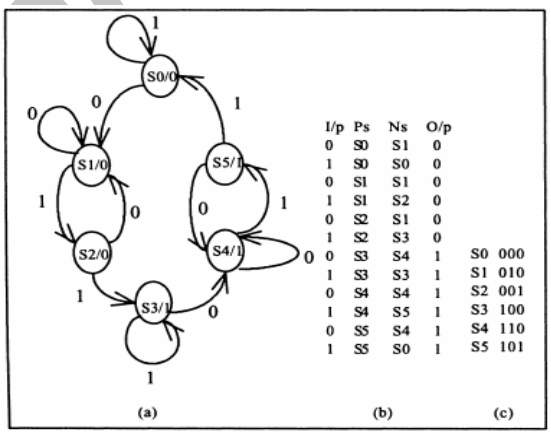
www.SID.ir

جدول۲: مقایسه نتایج بدست آمده برای مدار آشکار ساز دنباله ۱۰۱۰ توسط روش پیشنهادی با روش ارائه شده در [۱۰]

روش پیشنهادی توسط آقای آلمینی[۱۰]	روش پیشنهادی
$D_A=X'A'B+X'AB'+XAB$	$D_A=XB'+A$
$D_B=A'B+AB'+XB'$	$D_B=X$
$Z=X'AB'$	$Z=X'A'B$
زیرمدار A =۱۲ گیت	زیرمدار A =۲ گیت
زیر مدار B= ۲ گیت	زیر مدار B= ۳ گیت
زیر مدار C= ۲ فلیپ فلاپ	زیر مدار C= ۲ فلیپ فلاپ

ب) مدار منطقی ترتیبی با شش حالت میانی

در این بخش نیز یک مدار آشکار ساز دنباله دیگر توسط روش پیشنهادی، آزمایش شده است. این مدار دارای شش حالت میانی است و به همین جهت به منظور ساختن آن نیاز به سه فلیپ فلاپ می باشد. گراف انتقال حالت این مدار در شکل ۱۰نمایش داده شده است.



شکل۱۰: دیاگرام حالت مدار آشکار ساز دنباله [۸]

در این مثال نیز، این مدار را به روش توضیح داده شده در مثال قبل تکامل داده ایم و مدار تکامل یافته در شکل ۱۱نمایش داده شده است. همانطور که در این شکل مشاهده می نمایید، در مدار مورد نظر، در زیر مدار A، از هفت گیت استفاده شده است و این در حالی است که در زیر مدار B، از گیتی استفاده نشده است. در این آزمایش، ماکزیمم تعداد نسل ها به منظور تکامل زیر مدار ورودی A، ۲۰۲۵۰ نسل و به منظور تکامل زیر مدار خروجی B، ۱۲۱۵۰ نسل می باشد. این نتایج پس از ۵۰ بار اجرای الگوریتم بدست آمده اند.

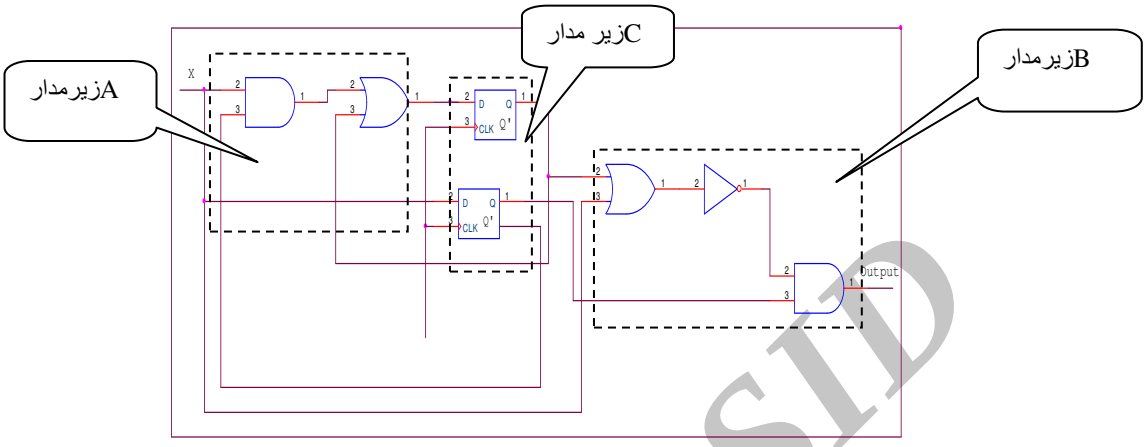
نتایج بدست آمده توسط روش پیشنهادی در مقایسه با طراحی توسط روش دستی و نیز روش ارائه شده در[۸]، در جدول ۳ نمایش داده شده است. همانطور که مشاهده می نمایید، مدار بدست آمده به شیوه طراحی دستی، تقریباً از دو برابر تعداد گیت های استفاده شده در مدار تکامل یافته با روش ما استفاده می نماید. همچنین روش پیشنهادی در [۸] نیز از تعداد گیت های بیشتری نسبت به روش ارائه شده توسط ما استفاده می نماید. ماکزیمم تعداد نسل های اجرای الگوریتم در [۸]، ۵۰۰۰۰ نسل می باشد، در حالی که در روش ما ماکزیمم تعداد نسل های اجرای الگوریتم ۲۰۲۵۰ نسل می باشد.

جدول۳: جدول مقایسه نتایج بدست آمده برای طراحی مدار آشکار ساز دنباله توسط روشهای ارائه شده

طراحی به روش دستی	روش پیشنهادی توسط خانم کالگانوا[۸]	روش پیشنهادی
$D_A=AC'+AX'+BCX'$	$D_A=XB$	$D_A=XB$
$D_B=BX+A'CX$	$D_B=X'$	$D_B=X'$
$D_C=BX+A'C'X'+A'B'X'+AC'X$	$Dc=XAC'+X'C+A'C$	$D_C=(XAC)'(C+XA)$
$Z=A+BC$	$Z=C$	$Z=C$
زیرمدار A =۱۷گیت	زیرمدار A =۸ گیت	زیرمدار A =۷ گیت
زیر مدار B= ۲گیت	زیر مدار B= ۰گیت	زیر مدار B= ۰ گیت
زیر مدار C= ۳ فلیپ فلاپ	زیر مدار C= ۳ فلیپ فلاپ	زیر مدار C= ۳ فلیپ فلاپ

Archive of SID

www.SID.ir



شکل۱۱: مدار تکامل یافته توسط روش پیشنهادی برای مدار آشکار ساز دنباله

۸ نتیجه گیری

این مقاله، روشی را به منظور طراحی و بهینه سازی مدارات ترتیبی سنکرون پیشنهاد داده است. در این روش، از الگوریتم ژنتیک به عنوان الگوریتم تکاملی استفاده شده است و مدارات مورد نظر در سطح گیت تکامل داده شده اند. نتایج بدست آمده توسط روش پیشنهادی، با دیگر روش های ارائه شده در [۸و۱۰] مقایسه شدند. این مقایسه ها نشان می دهند که روش ما قادر است تا در مدت زمان کمتر و با تعداد نسل های کمتری ، مدارات ترتیبی دیجیتال را تکامل دهد. همچنین، تعداد گیت های بکار رفته در مدار ترتیبی دیجیتال، توسط روش ما کاهش می یابد.

در کارهای آتی نیز می توان تلاش هایی را به منظور تکامل مدارات ترتیبی با مقیاس بزرگ تر که در صنعت کاربرد بیشتری دارند انجام داد.

مراجع

[1] X. Yao and T. Higuchi, "Promises and challenges of evolvable hardware," IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, Vol. 29, pp.87-97,1999.
[2] S. J. Mahdavi, "Evolutionary Design of Digital Reliable Circuits," Doctor of Philosophy in Electrical Engineering, Department of Electrical Engineering, Iran University of Science and Technology, Tehran, 2010.
[3] T. Higuchi, M. Murakawa, M. Iwata, I. Kajitani, W. Lia, and M. Salami "Evolvable Hardware at Function Level," in proceeding of int, pp.187-192, 1997.
[4] C. Manovit, C. Aporntewan and P. Chongstitvatana, "Synthesis of Synchronous Sequential Logic Circuits from Partial Input/Output Sequence", Vol.1478, pp.98-105,1998.
[5] C. Aporntewan, P. Chongstitvatana, "An On-Line Evolvable Hardware for Learning Finite-State Machine," in Proceeding of int, pp.13-15, 2000.
[6] A. T. Soliman, H.M. Abbas, "Synchronous Sequential Circuits Design using Evolutionary Algorithm," vol.4, pp.2013-2016, 2004.
[7] A. P. Shanthi, L. K. Singaram, "Evolution of Asynchronous Sequential Circuits," in Proceeding of the 2005 NASA/DoD Conference on Evolvable Hardwarex, pp.93-96, 2005.
[8] B. Ali, A. Almaini and T. Kalganova, "Evolutionary Algorithms and Their Use in the Design of Sequential Logic Circuits," in Genetic Program, Evolvable Machine, Vol.5, pp.11-29, 2004.
[9] http://www.Mathworks.com.
[10] A. E. A. Almaini, Electronic Logic Systems, Prentice-Hall, 3rd ED.1994,UK.
[11] E. Stomeo, T. Kalganova and C. Lambert, "Generalized Disjunction Decomposition for Evolvable Hardwar", Vol.36, No.2, pp.1024-1042, October 2006.

Archive of SID

www.SID.ir