# طراحی سخت افزار سیستم رمزگذاری و رمزگشائی *DES* با استفاده از زبان شبیه ساز *VHDL*

**احمد رضا شرافت**
استادیار بخش مهندسی برق - دانشکده فنی - دانشگاه تربیت مدرس
**محمود محضب**
دانشجوی کارشناسی ارشد بخش مهندسی برق - دانشکده فنی - دانشگاه تربیت مدرس
(تاریخ دریافت ۷۴/۱۲/۲۲، تاریخ تصویب ۷۸/۱۱/۲۳)

## چکیده

در این مقاله سخت افزار سیستم رمز نگار DES طراحی و عملکرد آن بررسی می شود. این سخت افزار که با پالس ساعت ۲۰MHz کار می کند ، قادر است که داده های ورودی را با نرخی برابر ۸۰ Mbps رمزکند که نسبت به سخت افزارهای موجود ، دوبرابر سریعتر است. سخت افزار طراحی شده بصورت مدار مجتمع ساخته می شود . کلیه مراحل طراحی و بررسی عملکرد سخت افزار با استفاده از زبان استاندارد VHDL انجام شده است. مزیت استفاده از VHDL این است که به سادگی می توان سخت افزار طر احی شده را در یک تراشه ساخت. این کار با استفاده از ابزار مبتنی بر VHDL انجام می شود.

**واژه های کلیدی** : رمزگذاری ، رمزگشائی ، سیستم DES ، زبان VHDL

## مقدمه

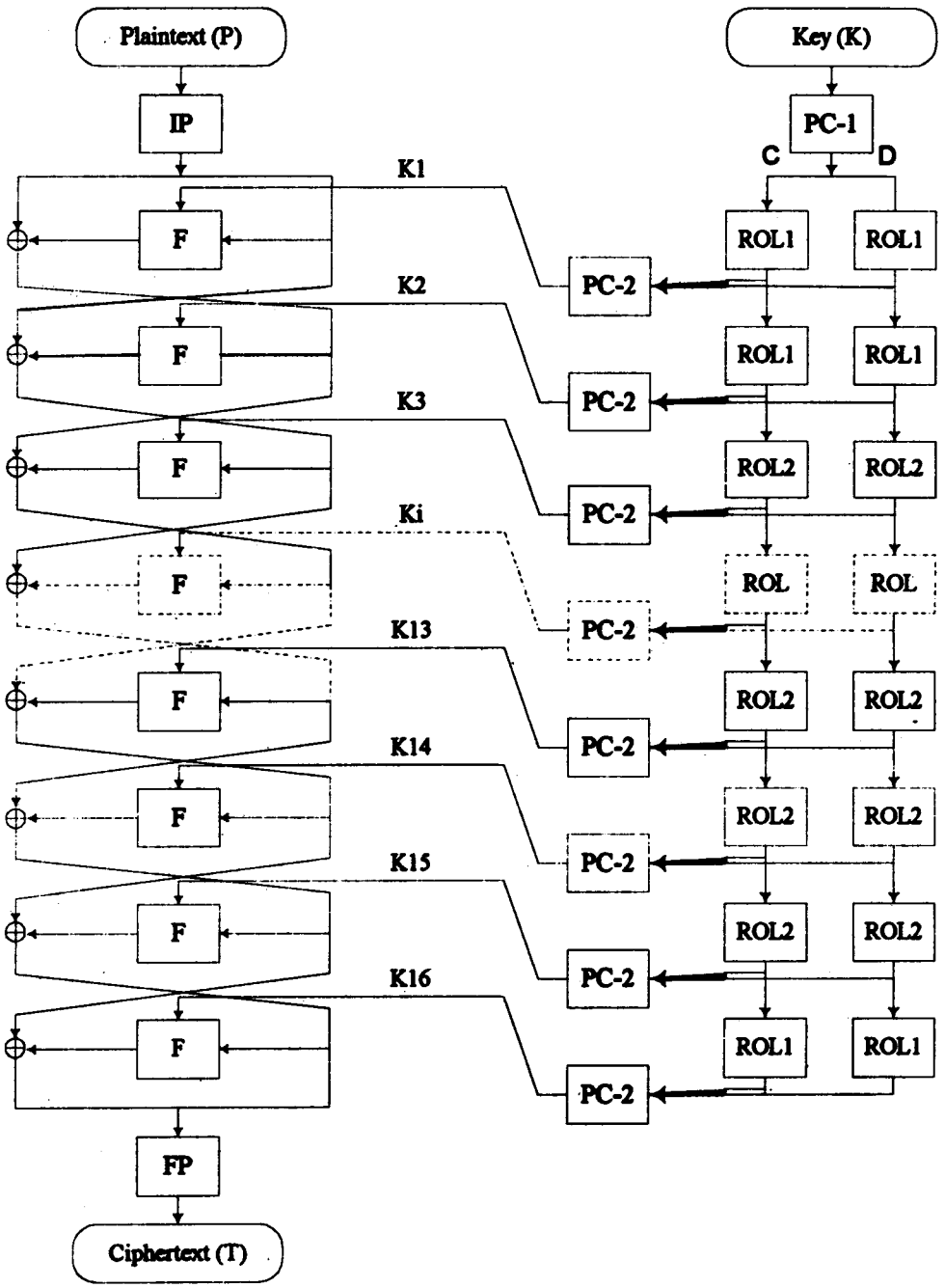
با گسترش اینترنت و افزایش استفاده از آن ، امکانات ارتباطی جدیدی بوجود آمده است و همراه با آن چالش هائی نیز در برابر ما قرار دارد. سادگی و مقرون به صرفه بودن استفاده از اینترنت آن را به ابزار بسیار مناسبی برای مبادله داده ها و اطلاعات تبدیل کرده است که از محیط مشترکی برای این کار بهره می جوید . کاربردهای آتی اینترنت ، بویژه در مورد تجارت الکترونیکی ، و مشترک بودن محیط انتقال داده ها ، مسئله امنیت اطلاعات را بعنوان یکی از مسائل کلیدی این زمینه مطرح ساخته است [۱].

امنیت اطلاعات با استفاده از سیستم های رمزنگاری محقق می شود. سیستم های رمزنگاری را می توان هم با استفاده از نرم افزار، و هم با بکارگیری سخت افزار ایجاد کرد. سیستم های نرم افزاری ، گرچه از قابلیت انعطاف بیشتری برخوردارند ، لکن در مقایسه با سخت افزار ، سرعت کمتری دارند. ارائه خدمات جدید چند رسانه ای در محیط اینترنت که سرعت آن بگونه ای قابل ملاحظه نسبت به خدمات سنتی (نظیر پست الکترونیکی و امثال آن ) بیشتر است ، بکارگیری سیستم های سخت افزاری را اجتناب ناپذیر کرده است. سخت افزار های رمزنگاری موجود ، سرعت انتقال اطلاعات را تا ۴ Mbytes/Sec فراهم می آورند [۲،۳] . این سرعت بخصوص برای خدماتی که اطلاعات بیشتری را در هر ثانیه جابجا می کنند کافی نیست .

الگوریتم بکاررفته در بسیاری از واحدهای رمزکننده ، سیستم استاندارد رمزگذاری داده ها یا DES است . این الگوریتم برای مراکز دولتی غیر نظامی توصیه شده است. انتخاب DES به عنوان استانداردی برای رمز کردن داده ها، سبب توجه بیشتر به این الگوریتم شده است.

DES یک سیستم رمزنگاری قالبی است که قالب های متن اصلی بطول ۶۴ بیت را با استفاده از یک کلید ۵۶ بیتی که ۸ بیت توازن نیز به آن اضافه می شود به قالب های ۶۴ بیتی متن رمز شده تبدیل می کند[ ۴] . نحوه عمل الگوریتم DES مطابق شکل (۱) به اختصار به این صورت است : ابتدا بیت های قالب ورودی توسط یک جایگشت اولیه ( IP ) بهم ریخته می شوند ، و سپس قالب

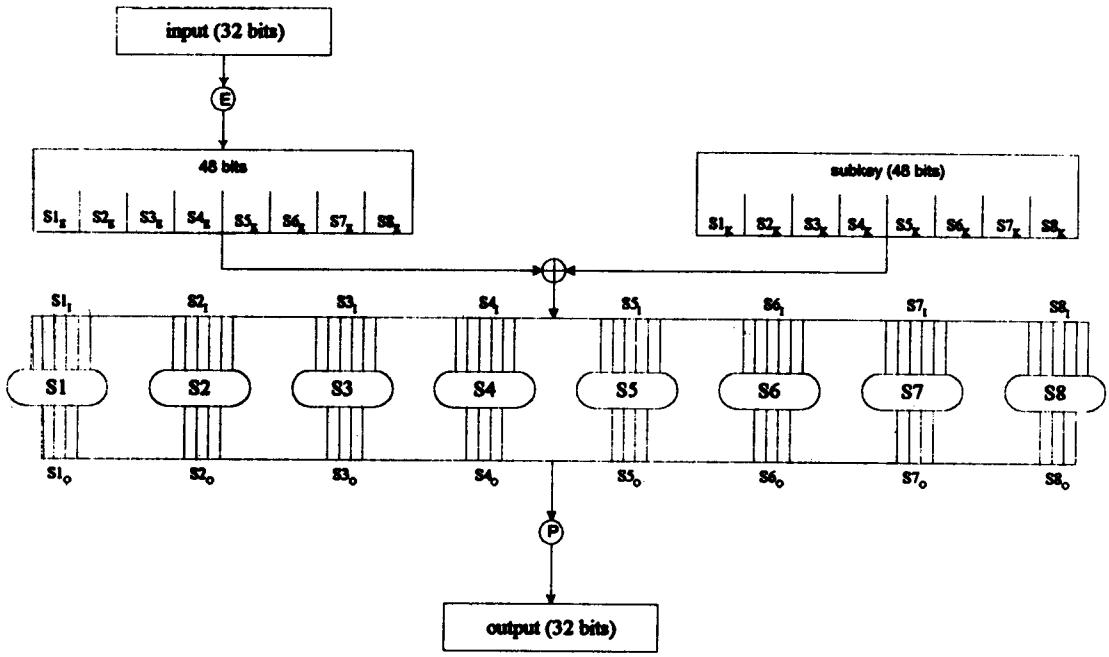
بهم ریخته شده به دو زیر قالب ۳۲ بیتی چپ و راست تقسیم می شود. آنگاه عملیات زیر طی ۱۶ دور متوالی بر دو زیر قالب چپ و راست اعمال می شود. در هر دور ، از یک کلید فرعی با طول ۴۸ بیت استفاده می شود . این کلید فرعی از روی کلید اصلی ۵۶ بیتی بصورتی که بعداً توضیح می دهیم ساخته می شود . کلیدهای فرعی متناظر با اولین ، دومین ، و ... شانزدهمین دور به ترتیب عبارتند از K1 ، K2 ، ... ، K16 . در هر دور، زیر قالب راست به همراه کلید فرعی مربوط به آن دور در ورودی تابع F قرار می گیرند و خروجی تابع F که یک قالب ۳۲ بیتی است با زیر قالب سمت چپ XOR می شود ( XOR قالب ها بیت به بیت انجام می پذیرد ) .




**شکل ۱ : الگوریتم رمزنگاری *DES* و الگوریتم تولید کلید آن [۲].**

بدین ترتیب زیر قالب جدید چپ تولید می شود و زیر قالب راست بدون تغییر باقی می ماند. سپس جای دو زیر قالب چپ و راست عوض می شود و عملیات دور بعدی آغاز می گردد. تعویض جای دو زیر قالب چپ و راست در دور شانزدهم انجام نمی شود. بعد از شانزده دور ، دو زیر قالب چپ و راست مجدداً در کنار همدیگر قرار می گیرند و یک قالب ۶۴ بیتی بوجود می آورند. آخرین مرحله ، اعمال یک جایگشت بر بیت های این قالب ۶۴ بیتی است که معکوس جایگشت اولیه (IP ) است و جایگشت نهائی (FP ) نامیده می شود. پس از اعمال این جایگشت، قالب ۶۴ بیتی متن رمز شده بدست می آید.

اکنون تابع F را توضیح می دهیم. مطابق شکل (۲) ، ابتدا




**شکل ۲ : تابع *F* در *DES* [۲]**

زیر قالب ۳۲ بیتی به ۴۸ بیت بسط داده می شود.

در این عمل که توسط تابع بسط دهنده E انجام می شود، ۱۶ بیت خاص در میان قالب ۳۲ بیتی تکرار می شوند. سپس ۴۸ بیت بدست آمده با ۴۸ بیت کلید فرعی XOR می شوند. نتیجه XOR به هشت دسته شش بیتی تقسیم می شود که هردسته در ورودی یکی از هشت S-box سیستم ( S1 ، S2 ، ... ، S8 ) قرار می گیرد. هر یک از این S-box ها دارای شش بیت ورودی و چهار بیت خروجی هستند ، یعنی یک نگاشت از فضای بردارهای شش بیتی به فضای بردارهای چهاربیتی انجام می شود. به این ترتیب ، خروجی هشت S-box تشکیل ۳۲ بیت می دهد که بعد از اعمال جایگشت P بر آنها ، خروجی تابع F بدست می آید.

نحوه عمل جایگشت اولیه IP ، جایگشت P و تابع بسط دهنده E در جدول های (۱) و(۲) و(۳) نشان داده شده است. در این جدولها ، ترتیب بیت ها بعد از انجام عمل نشان داده شده است. به عنوان مثال ، در جدول (۱) ملاحظه می کنیم که بیت شماره ۵۸ قالب ورودی جایگشت IP ، به اولین بیت قالب خروجی آن منتقل می شود.

**جدول ۱ : جایگشت اولیه *IP* [۵]**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

**جدول ۲ : جایگشت *P* [۵].**

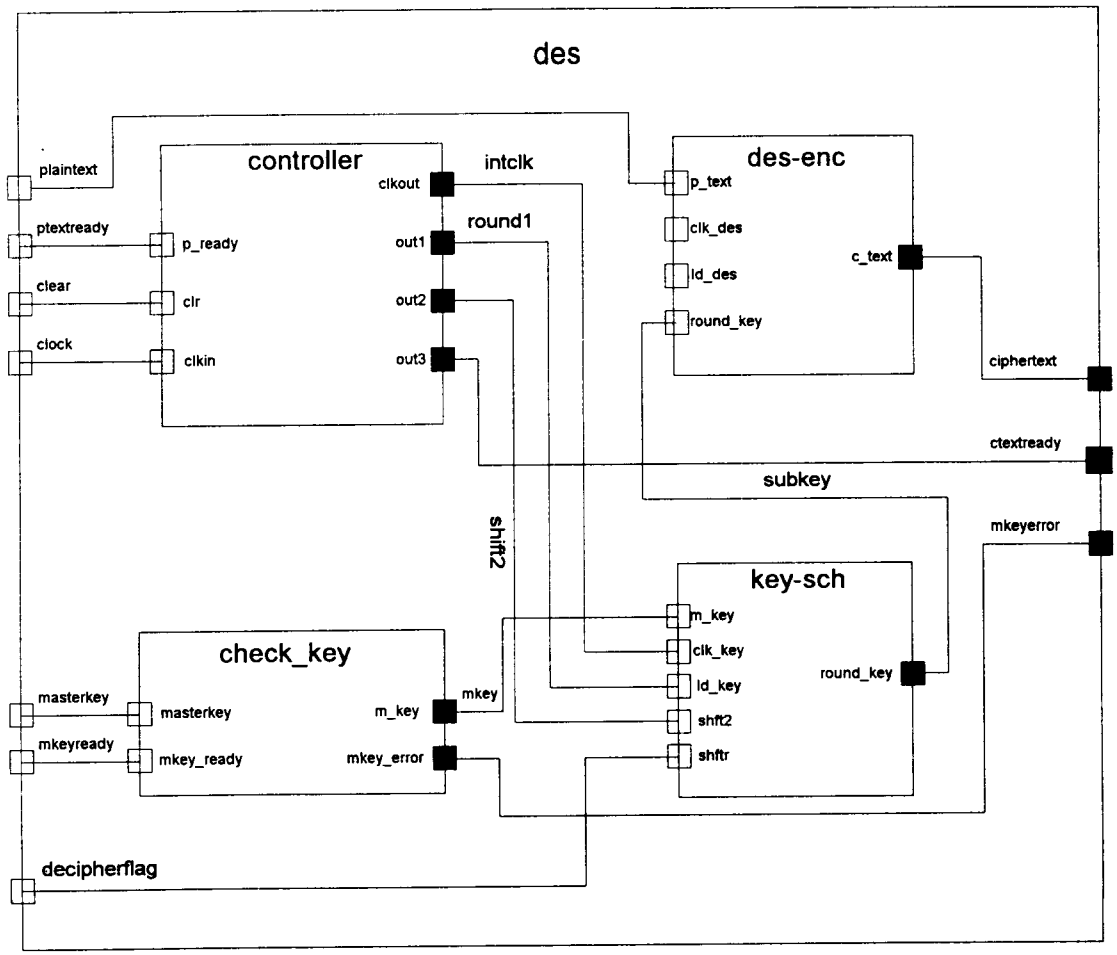
| | | | |
|---|---|---|---|
| 16 | 7 | 20 | 21 |
| 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 |
| 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 |
| 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 |
| 22 | 11 | 4 | 25 |

**جدول ۳ : بسط *E* [۵].**

| | | | | | |
|---|---|---|---|---|---|
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

بـــه ایـــن ترتیب ملاحظه می کنیم که رابطه بـــین قالب خروجی و قالب ورودی در این سیستم بسیار پیچیده است. این پیچیدگی بگونه ای است که هنوز هیچ حمله تحلیلـی موفق به شکستن این سیستم رمزنگار نشده است [۵، ۶] . به منظور تأمین ایمنی ارتباطات در شبکه های کامپیوتری بـــا ســـرعت زیـــاد ، در اختیـــار داشـــتن یـــک سخت افزار مناسب برای رمز کردن داده هاضروری اســت . اگر این سخت افزار بصورت مدار مجتمع در یک تراشه قرار گیرد، می تواند بگونه ای گسترده در شبکه مـورد اسـتفاده قرار گیرد. از جمله اینکه ، می تواند روی برد ســخت افـزار شبکه ( مربوط به لایه فیزیکی ) به عنوان بلوک رمزکننـده داده ها استفاده شود.

در ایـن مقالـه ، سـخت افـزار لازم را بـرای پیـاده سـازی الگوریتم DES ، طراحی می کنیم. انتخـاب DES ، بدلیـل کاربرد وسیع این سیستم در مصارف تجاری است که هنوز به عنوان یکی از پرقدرت تریـن سیسـتم هـای رمزنگـاری قالبی مطرح است .




**شکل ۳ : بلوک دیاگرام سخت افزار *DES***

برای طراحی سخت افزار DES از زبان VHDL استفاده کرده ایم . VHDL ، کاملترین و تواناترین زبان از نوع خود است که تقریباً در تمام مراکز علمی و صنعتی جهت طراحی سخت افزار بکار می رود. VHDL نه تنها از لحاظ قواعد برنامه نویسی ، غنی و انعطاف پذیر است ، بلکه توانایی توصیف عملیات همزمان و فرایندهای موازی را دراجزاء مدولها ، و سیستم های دیجیتال دارد. امروزه این زبان بصورت استانداردی برای طراحی سخت افزار تبدیل شده است [ ۷] .

## سخت افزار *DES*

در شکل (۳) بلوک دیاگرام سخت افزار DES نشان داده شده است. دراین شکل ، سخت افزار DES به چهار جزء مستقل هر یک با وظایف مشخص تقسیم شده است . بنابراین طراحی سخت افزار DES ، به طراحی چهار جزء نشان داده شده در شکل (۳) منجر می شود.

چهار واحد سخت افزار که در شکل (۳) نشان داده شده اند عبارتند از :

des-enc : وظیفه این واحد اجراء یک دور الگوریتم DES است.

key-sch : این واحد وظیفه تولید کلیدهای فرعی هر دور الگوریتم را برعهده دارد.

check-key : وظیفه این واحد ، بررسی توازن فرد بیت های کلید است . درصورتی که خطایی در توازن رخ دهد با فعال کردن یک سیگنال خطا ، آن را اعلام می کند.

controller : این واحد کنترل همهٔ سخت افزار را برعهده دارد.

برای اجراء ۱۶ دور الگوریتم DES ، بایستی که داده های ورودی ۱۶ مرتبه توسط واحد des-enc رمز شوند. در بخشهای بعدی ابتدا زمان بندی سیگنالهای داخلی و سپس طراحی هر یک از چهار واحد فوق را ارائه می دهیم. دیاگرام زمانی سیگنالها که در بخش بعدی نشان داده می شود تصویر واضح تری را از عملیات داخلی سخت افزار و وظایف هر یک از اجزاء ارائه می دهد و راهنمای خوبی برا ی طراحی ساختار داخلی هر یک از واحد هاست.

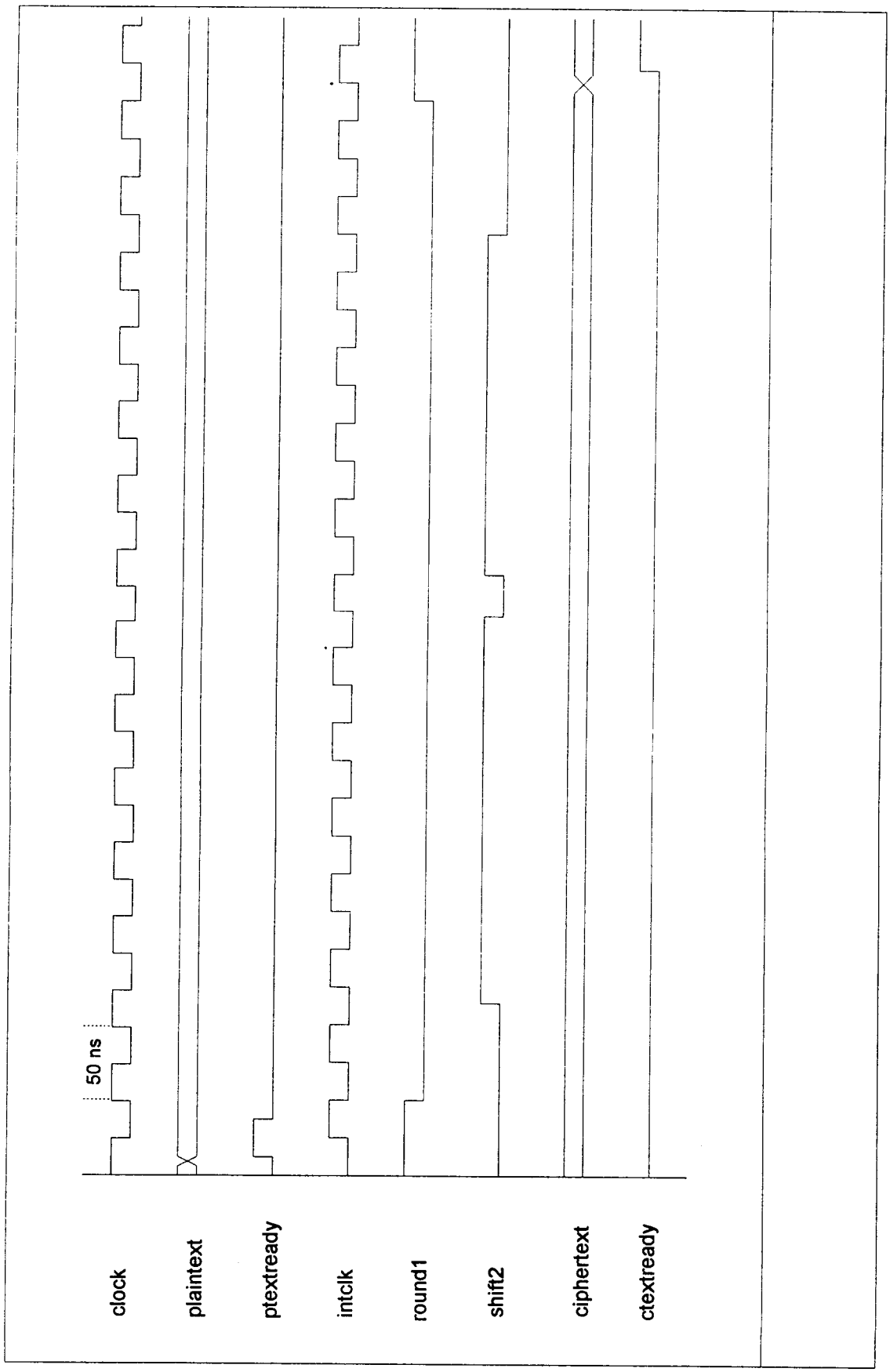
## زمان بندی سیگنالها

شکل (۴) وضعیت سیگنالهای ورودی ، خروجی ، و داخلی سخت افزار DES را نسبت به هم نشان می دهد. در اینجا سیگنال و اتصالات بین اجزا بصورت مترادف هم بکار می روند. اجزاء سخت افزار باید به گونه ای طراحی شوند که دیاگرام زمانی شکل (۴) را تولید کنند. نحوه کار مدار را می توان بصورت زیر تشریح کرد:

ابتدا کلید اصلی روی پورت masterkey گذاشته می شود و سیگنال mkeyready فعال می گردد ( این قسمت در شکل (۴) نشان داده نشده است ) . واحد بررسی کلید check-key ، توازن فرد هر بایت کلید را بررسی می کند (به عبارت دیگر ، در هر بایت کلید ، بایستی تعداد "۱" ها فرد باشد ). در صورتیکه توازن برقرار بود ، کلید اصلی به پورت خروجی منتقل می شود و روی سیگنال mkey قرار می گیرد. درغیر اینصورت فقط سیگنال اعلام خطا mkeyerror فعال می شود.

پس از انجام مقدمات اعمال کلید و آماده شدن آن ، ابتدا متن اصلی روی پورت plaintext گذاشته می شود و سیگنال ptextready فعال می گردد. با فعال شدن ptextready ، واحد controller ، ۱۶ پالس ساعت داخلی ( از روی پالس ساعت اصلی با ns ۵ تأخیر ) تولید می کند و بدین ترتیب واحدهای key-sch و des-enc به ترتیب شروع به تولید کلیدهای فرعی و انجام عملیات دوری الگوریتم می کنند. در هر پالس ساعت داخلی intclk ، یک دور الگوریتم به انجام می رسد. از طرفی واحد controller ، سیگنالهای round1 و shift2 را تولید می کند . فعال شدن سیگنال round1 برای هر دو واحد key-sch و des-enc مشخص می کند که در لبه صعودی پالس جاری بایستی عملیات اولین دور انجام گیرد. سیگنال shift2 برای واحد key-sch تعداد شیفت بیت های کلید را در هر دور مشخص می کند. صفر بودن shift2 بیانگر یک واحد شیفت و یک بودن آن بیانگر دو واحد شیفت در هر دور است. پس از شانزدهمین پالس ، اجراء الگوریتم کامل می شود و متن رمز شده روی پورت خروجی ciphertext گذاشته می شود. فعال شدن سیگنال ctextready آماده بودن متن رمز شده را اعلام می کند.

در اینجا لازم است که به سه نکته اشاره کنیم . نکته اول




**شکل ٤: دیاگرام زمانی سیگنالهای سخت افزار.**

اینکه کلیه عملیات در لبه صعودی پالس ساعت انجام می گیرد. همچنین واحدهای controller و check-key در لبه های صعودی پالسهای ptextready و mkeyready عملیات خود را آغاز می کنند.

نکته دوم اینکه عملیات نوشتن کلید اصلی و متن رمز شده روی پورتها و خواندن متن رمز شده از پورت خروجی ، مستقل از پالس ساعت است ( سیگنالهای mkerready ، ptextready و ctextready مستقل از وضعیت پالس ساعت اعمال می شوند ) . بنابراین در حین انجام عملیات رمزگذاری توسط سخت افزار می توان متن اصلی بعدی را آماده کرد و روی پورت ورودی نوشت . بدین ترتیب از سخت افزار استفاده بهینه به عمل آمده است.

آخرین نکته اینکه ، کلیه سیگنالهای استفاده شده در این سخت افزار و اجزاء داخلی آن ، دارای حالت فعال برابر یک هستند.

### واحد *check-key*

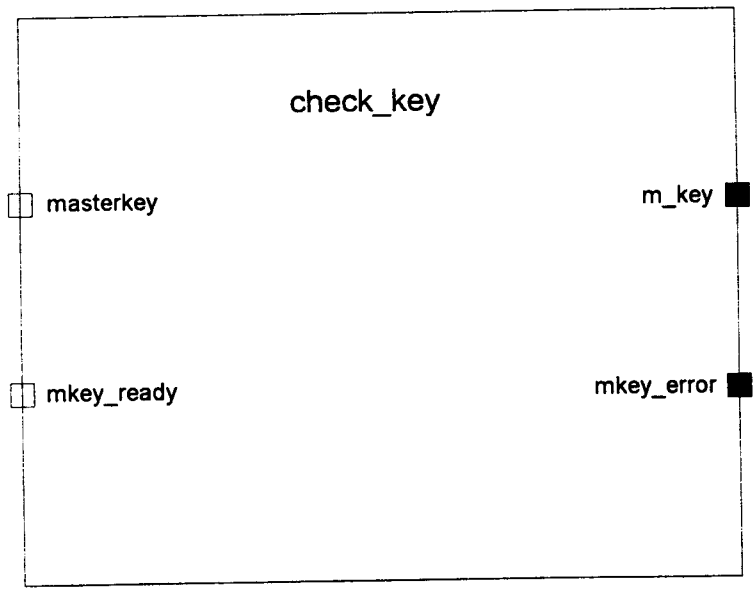
وظیفه این واحد بررسی توازن فرد هر یک از بایتهای کلید ورودی است. در DES هر کلید از هشت بایت تشکیل یافته ، که بیت هشتم در هر بایت ( بیت منتهی الیه سمت راست ) بیت توازن است . مقدار این بیت باید به گونه ای باشد که تعداد بیت های ' 1 ' در آن بایت ، عددی فرد باشد.

واحد check-key در شکل (۵) نشان داده شده است. نحوه عملکرد این واحد به شرح زیر است :

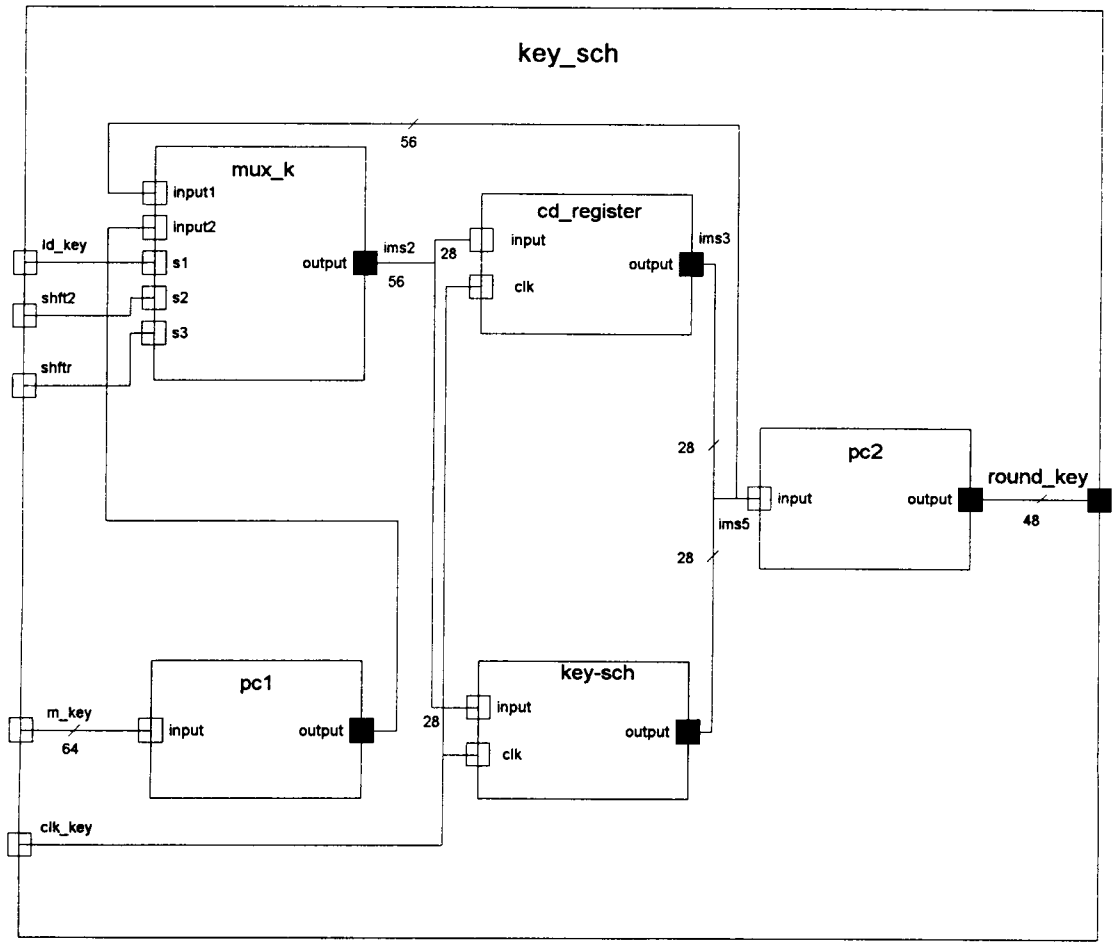
به محض فعال شدن سیگنال ورودی mkey-ready ، توازن کلید موجود روی پورت ورودی masterkey ، بررسی می شود. اگر حداقل یکی از بایتهای کلید دارای توازن فرد نباشد ، پس از یک تأخیر به میزان ۵ ns ، سیگنال خروجی mkey-error فعال می شود. با غیر فعال شدن سیگنال ورودی mkey-ready ، سیگنال خروجی mkey-error نیز پس از ۵ ns غیر فعال می شود. درصورتیکه تمام بایت های کلید ورودی دارای توازن فرد باشند ، این کلید پس از ۵ ns تأخیر ، به پورت خروجی m-key منتقل می شود و بدین ترتیب کلید اصلی، آماده استفاده در الگوریتم رمزنگاری خواهد بود.

### واحد *key-sch*

وظیفه این واحد ، تولید کلیدهای فرعی مناسب برای هر دور الگوریتم است . شکل (۶) ، بلوک دیاگرام واحد key-sch را نشان می دهد.




**شکل ۵ : واحد *check-key* .**




**شکل ٦ : واحد *key-sch***

ورودی های این جزء سخت افزار عبارتند از :

m-key : کلید اصلی برای رمزگذاری یا رمزگشایی از طریق این پورت وارد می شود.

clk-key : در هر دور ، رجیسترهای C و D از طریق این سیگنال پالس ساعت را دریافت می کنند.

ld-key : این سیگنال شاخص اولین دور الگوریتم است.

shft2 : اگر این سیگنال غیر فعال باشد ، دو نیمه کلید در هر دور یک بیت شیفت می یابند، در غیراینصورت با فعال شدن این سیگنال تعداد شیفت برابر دو بیت خواهد بود.

shftr : اگر این سیگنال غیرفعال باشد ، جهت شیفت به سمت چپ ( عملیات رمزگذاری ) ، و در غیراینصورت با فعال شدن این سیگنال ، جهت شیفت به سمت راست (عملیات رمزگشایی ) خواهد بود.

پورت خروجی این واحد ، round-key ( شامل ۴۸ بیت ) است که در هر دور حاوی کلید فرعی همان دور است .

همانگونه که از شکل (۶) پیداست ، واحد key-sch خود از پنج جزء سخت افزاری دیگر تشکیل یافته است. این اجزاء با اتصالات نشان داده شده در شکل (۶) ، وظیفه تولید کلیدهای هر دور را انجام می دهند. از آنجائیکه درهر لبه صعودی پالس ساعت یک دور الگوریتم انجام می شود ، بنابراین تأخیر این اجزاء باید به گونه ای باشد که در یک سیکل ساعت (فاصله دو لبه صعودی پالس ساعت که در فرکانس ۲۰ M H z برابر است با ۵۰ ns ) ، تمام این اجزاء وظایف خود را انجام داده باشند.

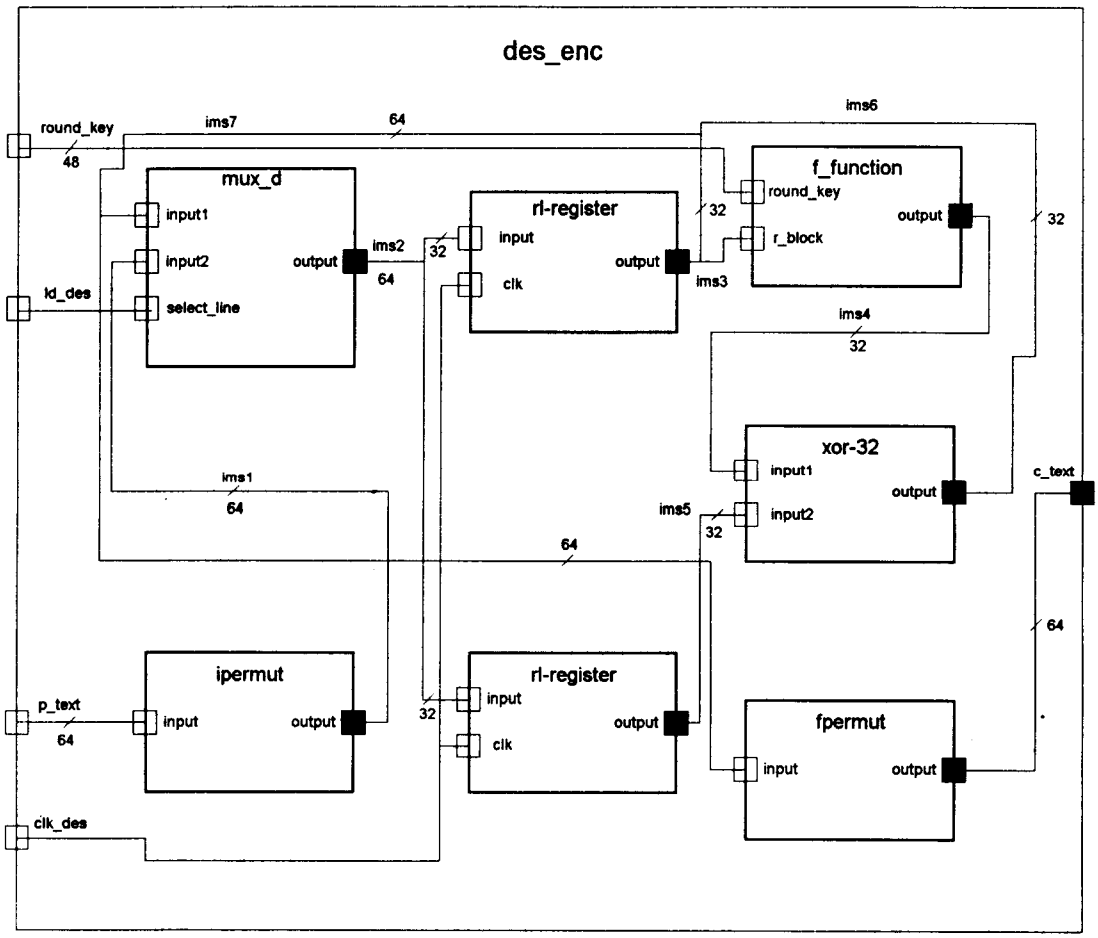
سیگنالهای میانی ims1 تا ims5 ، وظیفه اتصالات بین اجزاء را برعهده دارند. واحدهای pc1 و pc2 ، همان جایگشت های PC-1 و PC-2 را در الگوریتم تولید کلید ( شکل ۱ ) بوجود می آورند. ساختمان داخلی این واحدها بسیار ساده و متشکل از یک سیمبندی ساده متقاطع بین

ورودی و خروجی است . تأخیر ناشی از این اجزاء برابر صفر است.

در اولین دور که سیگنال ld-key فعال است ، واحد mux-k ورودی input2 خود را که همان سیگنال ims1 ( حامل کلید اصلی بعد از اعمال جایگشت PC-1 ) است انتخاب می کند. دونیمه چپ و راست بسته به وضعیت سیگنالهای shft2 و shftr ، بصورت گردشی شیفت می یابند و در ورودی رجیسترهای cd-register ( هر نیمه در ورودی یک رجیستر ) قرار می گیرند. در لبه صعودی پالس ساعت clk-key ، دو نیمه کلید در ورودی واحد pc2 قرار می گیرند و خروجی این واحد ، کلید فرعی این دور است . خروجی های رجیسترها ، بصورت فیدبک در دیگر ورودی واحد mux-k قرار می گیرند تا در دورهای بعدی استفاده شوند.

نکته اساسی در استفاده از دو رجیستر در این سخت افزار، جدا سازی دورهای متوالی الگوریتم است. زیرا واحد key-sch فقط یک دور الگوریتم تولید کلید را اجراء می کند و برای تولید هر شانزده کلید فرعی از همان سخت افزار استفاده می شود. وجود رجیسترهای cd-register باعث می شود که پس از اعمال پالس به این رجیسترها ، کلید دوربعدی در ورودی این رجیسترها باقی بماند تا پالس بعدی اعمال شود.

بدین ترتیب یک ساختار چرخشی زمان بندی شده (متناوب ، با دوره تناوبی برابر سیکل ساعت ) در تولید کلیدهای فرعی بوجود می آید.




**شکل ۷ : واحد *des-enc***

### واحد *des-enc*

وظیفه این واحد اجراء یک دور عملیات رمزگذاری در هر پالس ساعت است. شکل (۷) ، بلوک دیاگرام واحد des-enc را نشان می دهد . ورودی های این واحد عبارتند از :

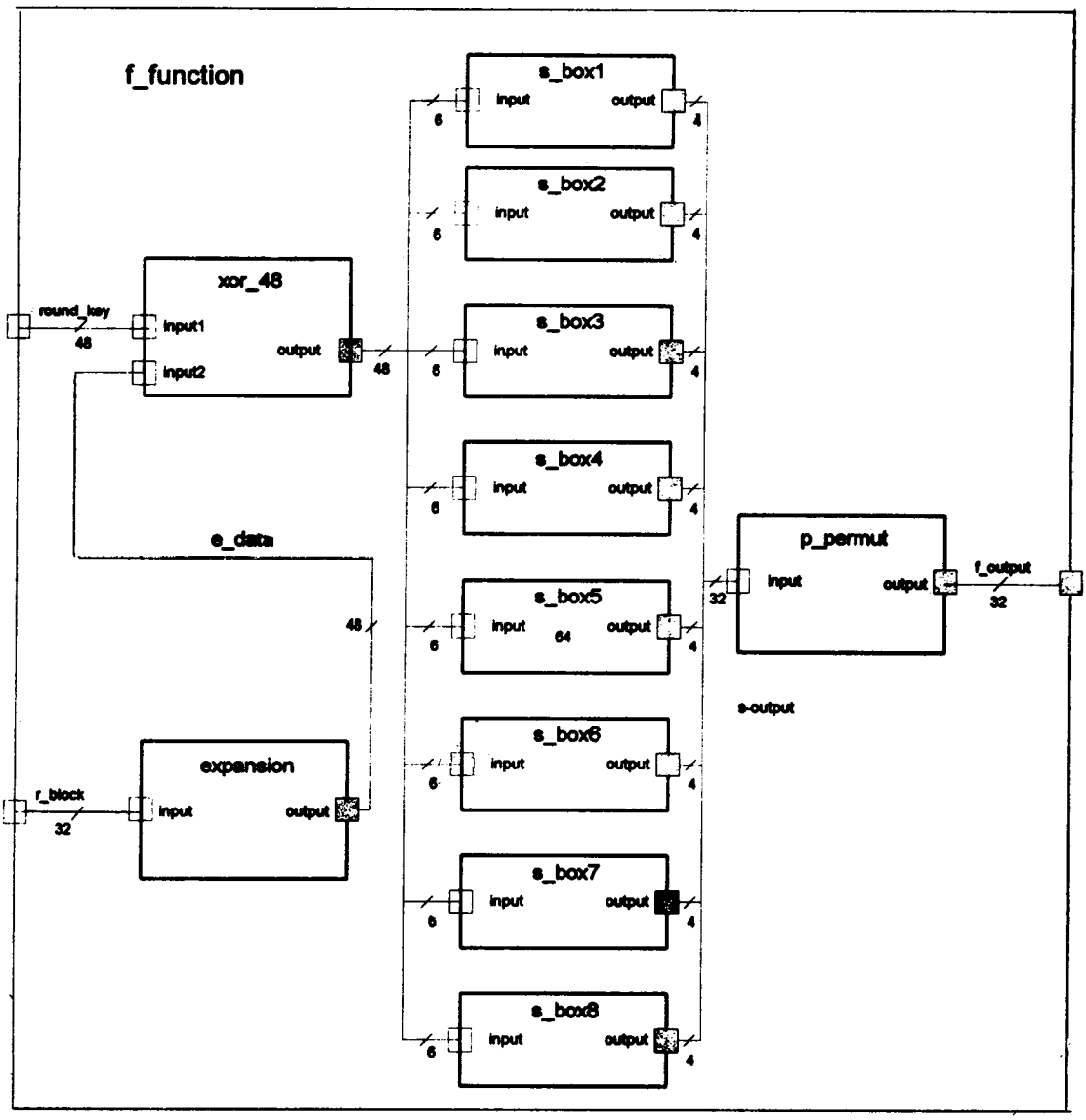
**p-text** : یک بردار باینری متشکل از ۶۴ بیت حاوی متن اصلی

**round-key** : یک بردار باینری متشکل از ۴۸ بیت حاوی کلید فرعی

**clk-des** : پالس ساعت رجیسترهای R و L

**ld-des** : سیگنال شاخص اولین دور الگوریتم پورت خروجی این واحد **c-text** ، یک بردار باینری متشکل از ۶۴ بیت است که پس از اتمام عملیات هر دور ، حاوی نتایج آن دور ( بعد از اعمال جایگشت نهایی ) است. بنابراین بعد از ۱۶ دور این پورت حاوی متن رمز شده خواهد بود.

همانگونه که از شکل (۷) پیداست ، واحد des-enc ، خود از هفت جزء سخت افزاری دیگر تشکیل یافته است. این اجزاء با اتصالات نشان داده شده در شکل (۸)، وظیفه اجراء عملیات یک دور الگوریتم را برعهده دارند. سیگنالهای میانی ims1 تا ims6 ، اتصالات بین اجزاء را برقرار می سازند. واحدهای ipermut و fpermut ، به ترتیب جایگشت های اولیه IP و نهایی FP را در الگوریتم رمزگذاری ( شکل ۱ ) بوجود می آورند. ساختمان داخلی این واحدها بسیار ساده و متشکل از یک سیمبندی ساده




شکل ۸ : واحد *f-function*

و متقاطع بین ورودی و خروجی است و بنابراین تأخیر ناشی از این اجراء صفر است. واحد mux-d در دور اول که سیگنال ld-des فعال است ، ورودی input2 ( متن اصلی بعد از جایگشت اولیه ) را ، و در دورهای بعدی، که سیگنال ld-des غیرفعال می شود ، ورودی دیگر خود را یعنی input1 ، که جای دو نیمه چپ وراست آن عوض شده ، برای خروجی انتخاب می کند. میزان تأخیر ناشی از واحد mux-d برابر ۵ ns است.واحد rl-register دقیقاً مانند واحد cd-register ( شکل ۷ ) عمل می کند. این رجیسترها زمان بندی اجراء هر دور را برعهده دارند. هر دو واحد des-enc و key-sch از یک منبع، پالس ساعت را دریافت می کنند و بدین ترتیب یک همزمانی بین این دو واحد بوجود می آید.

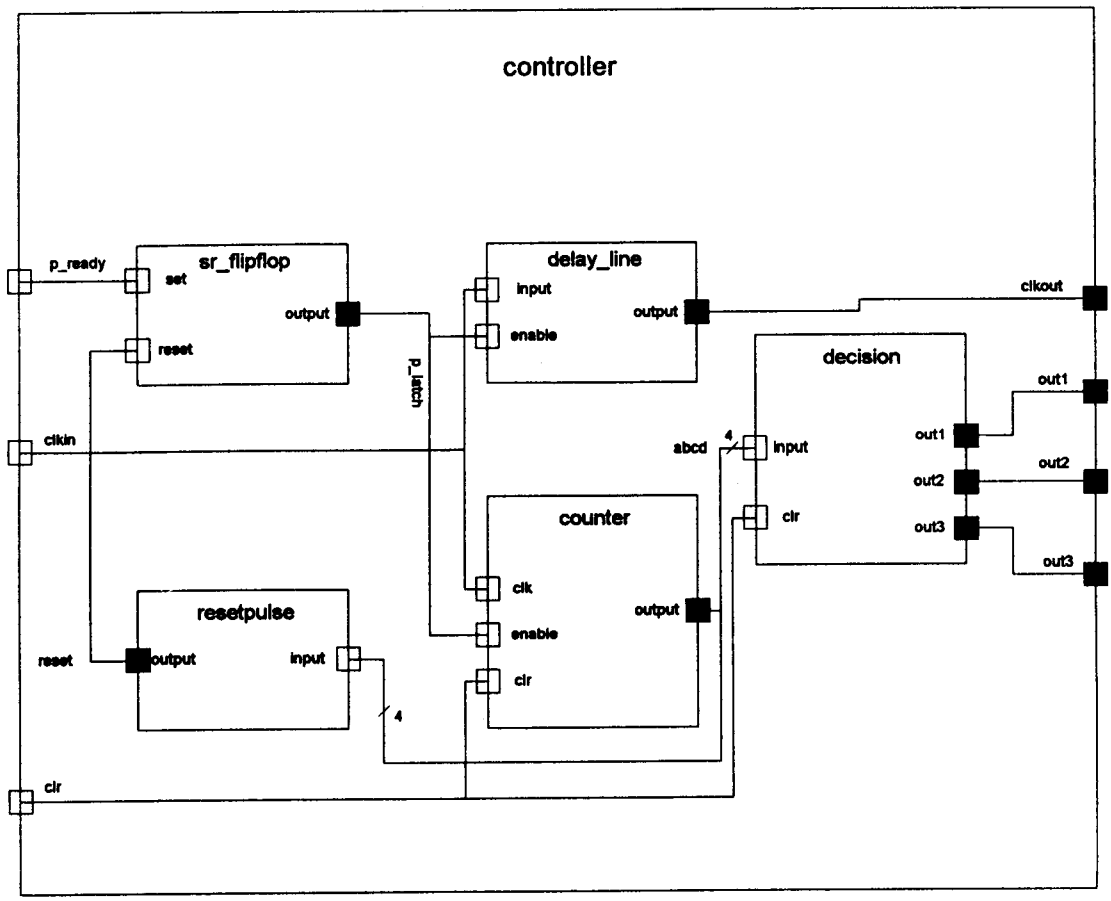
واحد xor-32 ، عمل XOR دو ورودی خود را انجام می دهد . هر کدام از ورودی ها شامل ۳۲ بیت است و XOR آنها بصورت بیت به بیت انجام می شود. میزان تأخیر ناشی از این واحد برابر ۶ ns است . این واحد را می توان بصورت آرایه ای از ۳۲ دروازه XOR ، هر کدام با دو بیت ورودی ویک بیت خروجی ، تلقی کرد.

وظیفه واحد f-function ، اجراء عملیات تابع F ( شکل ۲ ) در الگوریتم DES است که در بخش بعدی شرح ساختار داخلی آن ارائه می شود.

### واحد *f-function*

شکل (۸) ، واحد f-function و اجزاء آن را نشان می دهد.

واحدهای expansion و p-permut ، به ترتیب وظیفه اجراء توابع بسط E و جایگشت P در تابع F ( شکل ۲ ) را برعهده دارند. این واحدها نیز یک سیمبندی ساده و متقاطع بین ورودی و خروجی خود هستند و هیچگونه تأخیری در سیگنالها بوجود نمی آورند.




**شکل ۹ : واحد *controller***

واحد xor-48 ، همانند واحد xor-32 عمل می کند ، با این تفاوت که ورودی ها و خروجی این واحد هر کدام ۴۸ بیت هستند. میزان تأخیر ناشی از این واحد ns برابر ۶ است .
هشت S-box بکار رفته در واحد f-function ، بصورت هشت حافظه ROM طراحی شده اند. این حافظه های ROM حاوی ۶۴ کلمه چهاربیتی هستند. یعنی شش خط آدرس و چهار خط خروجی دارند . میزان تأخیر ناشی از S-box ها برابر ns ۱۱ است.

### واحد *controller*

کنترل مجموعه سخت افزار برعهده واحد controller است. این واحد ، پالس ساعت داخلی و دیگر سیگنالهای کنترلی را ، برای درست کار کردن دیگر واحد ها ، تولید می کند. این سیگنالها طبق دیاگرام زمانی شکل (۴) تولید می شوند. شکل (۹) ، بلوک دیاگرام سخت افزار واحد controller را نشان می دهد.
در شکل (۹) ، مشخص است که واحد controller از پنج جزء سخت افزاری دیگر تشکیل شده است. زمانی که p-ready فعال می شود ( هنگامیکه متن اصلی آماده رمزگذاری شد ، p-ready به مدت ۵۰ ns فعال می شود) ، در لبه صعودی پالس p-ready ، خروجی واحد sr-flipflop (p-latch ) نیز فعال می شود. فعال شدن p-latch باعث می شود که واحدهای counter و delay-line فعالیت خود را شروع کنند. counter یک شمارنده چهاربیتی است . این واحد به عنوان شمارنده دور بکار می رود. درهر لبه صعودی پالس ورودی clkin ، یک واحد به مقدار شمارنده اضافه می شود. پس از اعمال ۱۶ پالس ساعت ، واحد resetpulse ، فعال می شود و با تولید پالس reset ، واحد sr-flipflop را غیر فعال می کند. بدین ترتیب ، کار واحدهای counter و delay-line متوقف می شود. در خلال این ۱۶ پالس ، واحد delay-line ، ۱۶ پالس داخلی از روی پالس ورودی و با کمی تأخیر بوجود می آورد. این ۱۶ پالس برای انجام ۱۶ دور رمزنگاری مورد استفاده قرار می گیرند. همچنین واحد decision ، براساس عدد موجود در شمارنده خروجی های مناسب را برای این واحد ( شکل۴ ) تولید می کند.
ورودی clr در واحد controller برای ایجاد شرایط اولیه مناسب در این سخت افزار تعبیه شده است. این سیگنال فقط در ابتدای کار سخت افزار و به مدت ۱۰۰ ns فعال می شود.

## نتایج

عملکرد سخت افزاری که به این ترتیب طراحی ، و ساختار و اجزاء آن بیان شد ، را با استفاده از زبان شبیه ساز VHDL مورد بررسی و تجزیه و تحلیل قرار دادیم. سخت افزار طراحی شده DES قادراست که با پالس ساعت ۲۰MHz ، داده های ورودی را با نرخی برابر ۸۰ Mbps رمز کند. این نرخ رمزنگاری بویژه برای کاربردهای شبکه های انتقال اطلاعات که از فن آوری ماهواره ای استفاده می کنند بسیار مناسب است ، و حتی برای سیستم های نسل آتی ارتباطات ماهواره ای نیز کاربرد دارد. از خصوصیات دیگر این سخت افزار این است که کنترل آن کاملاً بصورت سخت افزاری انجام می شود. بدین ترتیب بازدهی این سخت افزار نسبت به سخت افزارهای موجود DES [۳] بیش از دو برابر می شود.
طراحی و شبیه سازی سخت افزار DES با استفاده از زبان VHDL و امکانات آن انجام شده است. نرم افزار مورد استفاده در این طراحی Ver.3.2 V_SYSTEM است که تحت WINDOWS اجرا می شود. این نرم افزار محصول شرکت Model Technology است و امکانات بسیار قوی برای شبیه سازی سخت افزار های طراحی شده دارد.
برای آزمایش سخت افزار و اطمینان از صحت عملکرد آن بایستی ابتدا تعدادی متن اصلی و متن رمز شده متناظر آن تحت کلیدهای خاصی تولید شوند. برای تولید این داده ها از DES نرم افزاری استفاده شده است. متن های اصلی و متن های رمز شده متناظر ، تحت کلیدهای خاص در جدول (۴) برای عملیات رمزگذاری و در جدول (۵) برای عملیات رمزگشایی ارائه شده اند. در این جداول اعداد در مبنای ۱۶ درج شده اند. ذکر این نکته ضروری است که امکان آزمایش عملکرد سخت افزار طراحی شده برای تمام قالب های ورودی و تمامی کلیدهای ممکن وجود ندارد ، زیرا این بررسی نیاز به $2^{56} \times 2^{64}$ عمل رمزگذاری و همین تعداد عمل رمزگشائی دارد که از نظر زمانی غیر ممکن است . با این حال ، داده های آزمایشی جدول های (۴) و (۵) ، ما را از صحت طراحی مطمئن می سازد. زیرا

در رمزنگار DES به علت خاصیت بهمنی [ ۹ ] ، هر یک از بیت های متن رمز شده به تمام بیت های متن اصلی و کلید وابسته است . بطور معکوس نیز ، تغییر هر یک از بیت های متن اصلی و کلید روی تمام بیت های متن رمز شده اثر می گذارد ، به گونه ای که تغییرات بسیار کوچک در متن اصلی یا کلید ، منجر به تغییرات بزرگ در متن رمز شده می شود. بنابراین اگر سخت افزار طراحی شده حتی دارای اشکال جزئی در طراحی باشد ، به علت خاصیت بهمنی ، نتایج خروجی بسیار دور از انتظار خواهند بود. به عبارت دیگر اگر در عملیات یک دور ، یک بیت خطا رخ دهد ، این یک بیت در دورهای بعدی منجر به انتشار خطا ، و در نتیجه پس از ۱۶ دور ، تغییرات کلی در متن رمز شده می شود. بنابراین ، مطمئن هستیم که داده های جدول های (۴) و (۵) برای آزمایش سخت افزار کافی هستند.

برای اعمال ورودیها به سخت افزار یک واحد آزماینده ساده طراحی شده است . برای شروع کارسخت افزار بایستی که ابتدا پالس ساعت را اعمال و سپس سیگنالهای ptexready و clear را فعال کرد .فعال شدن clear باعث می شود که شرایط اولیه مطلوب در سخت افزار ایجاد شود و برای اجرای عملیات رمزگذاری یا رمزگشایی آماده گردد. طول سیگنال clear بایستی حداقل ۱۰۰ ns باشد . برای مشاهده نحوه عملکرد سخت افزار و مشاهده نتایج خروجی می توان از امکانات نرم افزار V_SYSTEM استفاده کرد .

در سخت افزار طراحی شدهDES فقط سخت افزار یک دور این الگوریتم پیاده سازی شده است و برای اجرای ۱۶ دور الگوریتم از همین سخت افزار استفاده می شود. بدین ترتیب که در هر پالس ساعت عملیات یک دور انجام می شود. در نتیجه برای اجرای کامل الگوریتم، تعداد ۱۶ پالس ساعت مورد نیاز است. با بهره گیری از روش خط لوله می توان سرعت سخت افزار را ۱۶ برابر کرد [۸ ] . در این صورت در هر پالس ساعت یک قالب متن رمز شده تولید می شود و نرخ ورودی - خروجی داده هابه ۱/۲۸ Gbps می رسد.

**جدول ٤ : داده های آزمایشی برای رمزگذاری.**

| *Key* | *Plaintext* | | *Ciphertext* | |
|---|---|---|---|---|
| ababab<br>ababab | aaaaaaaa | aaaaaaaa | c4322be1 | 9e9a5a17 |
| | f0f0f0f0 | f0f0f0f0 | abff1b2d | 4671fa32 |
| | 00000000 | 00000000 | c33f4517 | dd950a2e |
| | ffffffff | ffffffff | f32823b6 | fc574774 |
| fefefefe<br>fefefefe | aaaaaaaa | aaaaaaaa | 4ef6027f | c14d2fa1 |
| | f0f0f0f0 | f0f0f0f0 | 230da379 | e0d5b321 |
| | 00000000 | 00000000 | caaaaf4d | eaf1dbae |
| | ffffffff | ffffffff | 7359b216 | 3e4edc58 |
| 01010101<br>01010101 | aaaaaaaa | aaaaaaaa | 3ae71695 | 4dc04e25 |
| | f0f0f0f0 | f0f0f0f0 | 6d7d7df6 | 9335c6c2 |
| | 00000000 | 00000000 | 8ca64de9 | c1b123a7 |
| | ffffffff | ffffffff | 355550b2 | 150e2451 |

**جدول ۵ : داده های آزمایشی برای رمز گشایی .**

| *Key* | *Ciphertext* | *Plaintext* |
|---|---|---|
| abababab<br>abababab | a a a a a a a a a a a a a a a a | d 8 4 2 1 5 3 8 7 b 7 f 9 e 3 d |
| | f 0 f 0 f 0 f 0 f 0 f 0 f 0 f 0 | d 2 7 f 7 5 7 7 7 6 e f e e a b |
| | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | d b 4 9 c 3 3 6 9 3 4 9 4 a 5 6 |
| | f f f f f f f f f f f f f f f f | c c c 1 1 e 4 2 5 9 b 0 2 2 c 7 |
| fefefefe<br>fefefefe | a a a a a a a a a a a a a a a a | 4 e f 6 0 2 7 f c 1 4 d 2 f a 1 |
| | f 0 f 0 f 0 f 0 f 0 f 0 f 0 f 0 | 2 3 0 d a 3 7 9 e 0 d 5 b 3 2 1 |
| | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c a a a a f 4 d e a f 1 d b a e |
| | f f f f f f f f f f f f f f f f | 7 3 5 9 b 2 1 6 3 e 4 e d c 5 8 |
| 0 1 0 1 0 1 0 1<br>0 1 0 1 0 1 0 1 | a a a a a a a a a a a a a a a a | 3 a e 7 1 6 9 5 4 d c 0 4 e 2 5 |
| | f 0 f 0 f 0 f 0 f 0 f 0 f 0 f 0 | 6 d 7 d 7 d f 6 9 3 3 5 c 6 c 2 |
| | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 8 c a 6 4 d e 9 c 1 b 1 2 3 a 7 |
| | f f f f f f f f f f f f f f f f | 3 5 5 5 5 0 b 2 1 5 0 e 2 4 5 1 |

## مراجع


1 - Zhang, Z. (1997). "Crypography and electrons; commerce ." *Technical Report, Virginia Tech.*

2 - Teltrend Reference Manual. (1997). *The compression and encryption facility* . Hampshire, UK.

3 - Tundra Semiconductor Corporation Products (1999). CA20C03 A/W DES Encryption Processor. Ontario, Canada.

4 - National Bureau of Standards. (1997). *Data encryption standard,* FIPS PUB 46 Washington DC, 15 January.

5 - Biham, E. and Shamir, A. (1993). *Differential cryptonalysis of the data encryption standard.* Springer-Verlag, New York.

6 - Schneier, B. (1996). *Applied crypotography.* John Wiley and Sons, Inc.

7 - Navabi, Z. (1993). *VHDL analysis and modeling of digital systems.* McGraw-Hill.

8 - Fairfield, R. C., Matusevich, A. and Plany, J. "An LSI digital encryption processor (DEP). " *Advances in Cryptology: Proc. of CRYPTO '84,* Springer-Verlag, Vol. 196, PP. 115-143.

۹ - هندسی ، ف. " نقد و بررسی رمزنگار DES ." پایان نامه کارشناسی ارشد دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان ، (۱۳۶۸).