

به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر و فناوری اطلاعات



فاز دوم پروژه

درس معماری کامپیوتر

اعضای گروه:

سید محمد هادی حسینی

حسین محمدی

اریا خلیق

استاد درس: دکتر حسینی منزه

تدریسار درس: آقای محرابی

نیم سال اول 1397-98

1. توضیحات ابتدایی انتخاب (suitable Function) :

ما در فاز یک پروژه به گلوگاه در عمل تقسیم و تکرار بسیار زیاد آن رسیدیم به همین دلیل تقسیم اعداد بدون علامت به صورت restoring را انتخاب کردیم و حالت ۳۲ بیتی آن را پیاده‌سازی کردیم و ما این پیاده‌سازی را به کمک ise انجام دادیم .

2. طراحی مدار تقسیم کننده مربوطه :

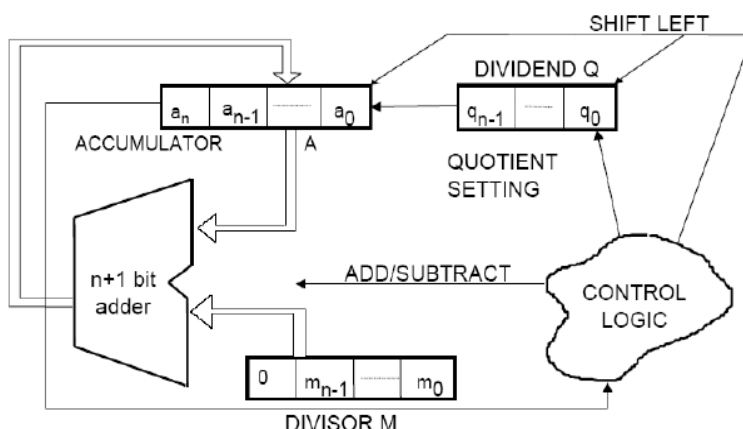
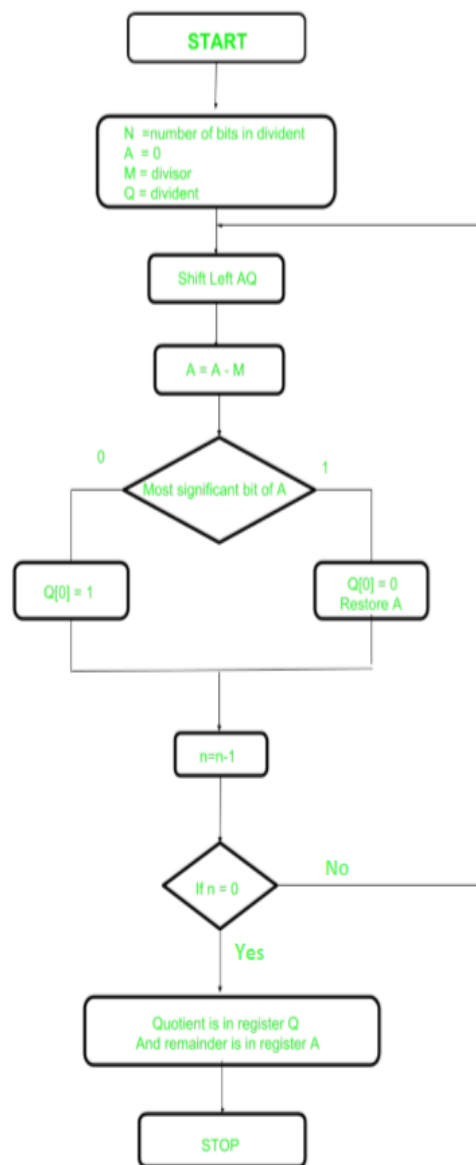


Fig. 1 Hardware Design of Restoring Division Algorithm.

3. توضیح کلی روند انجام کار از طراحی تا پیاده سازی :

هدف کل برنامه پیاده‌سازی یک تقسیم کننده ۳۲ بیتی میباشد که بتواند با کمترین تأخیر و منابع ، خواسته ما رو برآورده کند و ما برای انجام این کار از الگوریتم تقسیم به صورت restoring استفاده کردیم و برای نوشتن کد برنامه و تست آن و گرفتن فایلی که منابع مصرفی و تأخیر رو نشون بدهد از ise استفاده کردیم و روند کلی کار در الگوریتم restoring به این صورت میباشد که شکل طراحی اش در بالا آمده است و به این صورت کار میکند که ما به تعداد بیت‌های مقسوم یک فرایندی را انجام میدهم و در هر مرحله ابتدا مقسوم علیه را یک شیفت به چپ میدهم و سپس از عدد مون کم میکنیم و سپس بیت اول مقسوم علیه را در نظر میگیریم و اگر صفر باشد به این معناست که قابل تقسیم نمیشد و باید دوباره اون مقداری را که کم کردیم رو اضافه کنیم و در آخر هم میریم مرحله بعد . که صورت این الگوریتم به صورت فلوچارت در پایین آورده شده است و در قسمت بعد هم به صورت مختصر راجب نحوه پیاده سازیش نوشتیم .

4. فلوجارت الگوریتم پیاده‌سازی تقسیم با روش (Restoring) :



5. توضیحاتی در مورد پیاده سازی کد :

با استفاده از فلوچارت الگوریتم FSM را رسم کردیم به گونه‌ای که هر Block در فلوچارت را برابر یک State در FSM گرفتیم و این State ها با توجه به کلاک و شروط if در هر کلاک به یک State دیگر می‌روند .

6. استیت های پیاده‌سازی عبارت اند از :

```
shift_left_prime  
A_minus_M  
shift_left  
restore_A  
initial_state  
split_AQ  
assign_Q0  
decrease_N  
check_if_N_is_ZERO  
finish_state
```

که مطابق با فلوچارت الگوریتم کشیده و نامگذاری شده‌اند .

7. توضیحات فایل‌های ضمیمه شده :

فایل Restoring_Division.vhd که کد اصلی برنامه را دارا می‌باشد و کدی سنتزپذیر است که از نتیجه سنتز آن توانستیم فایل Delay and Area.txt را بدست بیاوریم.

فایل Test_Bench.vhd که تست بنچ برنامه می‌باشد که به کمک آن توانستیم صحت کدمون رو بررسی کنیم و ما این روند بررسی کد را در ise انجام دادیم . (تست بنچ : شامل کدی می‌باشد که برای کد اصلی برنامه مون نوشته میشود و ما در آن ورودی های کد اصلی رومون رو مقدار دهی میکنیم و سپس خود ide که در اینجا ise می‌باشد براساس این مقدار دهی میاد خروجی مرتبط با آن را به ما میدهد که به کمک آن میتوانیم صحت برنامه ای را که پیاده سازی کره ایم را بررسی کنیم).

فایل Delay and Area.txt برای تحلیل تأخیر و منابع مورد استفاده می‌باشد و این فایل را در ise بدست آوردیم و از نتیجه سنتز پذیر بودن کدمون حاصل شده است .