



در عملیات تقسیم یک مقسوم (Dividend) به نام x بر یک مقسوم علیه (Divisor) به نام d تقسیم می‌کنیم و یک خارج‌قسمت (Quotient) به نام q و یک باقیمانده (Remainder) به نام w به دست می‌آید که بین این مقادیر رابطه $x = q.d + w$ برقرار است، به گونه‌ای که باقیمانده با مقسوم هم‌علامت است و قدرمطلق باقیمانده از قدرمطلق مقسوم علیه کوچک‌تر است، یعنی $|w| < |d|$. برای سادگی در اینجا فقط عملیات تقسیم اعداد صحیح و مثبت را بررسی می‌کنیم، به عبارت دیگر مقسوم مثبت را بر مقسوم علیه مثبت تقسیم می‌کنیم و خارج‌قسمت و باقیمانده مثبت به دست می‌آید. فرض می‌کنیم مقسوم $2n$ بیتی، مقسوم علیه، خارج قسمت و باقیمانده همه n بیتی هستند. در این حالت اگر مقسوم علیه 0 باشد، عملیات تقسیم پاسخ ندارد (تقسیم بر صفر Division by Zero). اگر مقسوم علیه از n بیت پرارزش مقسوم کوچک‌تر یا مساوی آن باشد، خارج قسمت در n بیت جای نمی‌گیرد و اصطلاحاً می‌گوییم سرریز رخ داده است و عملیات تقسیم پاسخ ندارد. بنابراین شرط عدم بروز سرریز در تقسیم این است که مقسوم علیه از n بیت پرارزش مقسوم بزرگ‌تر باشد. برای انجام عملیات تقسیم دو روش کلی وجود دارد که عبارتند از: تقسیم با بازیابی (Restoring Division) و تقسیم بدون بازیابی (Non-Restoring Division).

تقسیم با بازیابی

الگوریتم تقسیم با بازیابی به این صورت است که در ابتدا باقیمانده پاره‌ای (Partial Remainder) با مقدار مقسوم، مقداردهی اولیه می‌شود. سپس در هر مرحله مقسوم علیه را از n بیت پرارزش باقیمانده پاره‌ای کم می‌کنیم (مکمل 2 مقسوم را با n بیت پرارزش باقیمانده پاره‌ای جمع می‌کنیم). اگر حاصل مثبت باشد بیت متناظر از خارج قسمت را 1 و اگر حاصل منفی باشد بیت متناظر از خارج قسمت را 0 می‌کنیم و مقسوم را با باقیمانده پاره‌ای جمع می‌کنیم تا مقدار قبلی آن بازیابی شود. در حقیقت در این روش باقیمانده پاره‌ای منفی برای مراحل میانی نیز پذیرفته نیست و بلافاصله به تصحیح آن اقدام می‌کنیم. الگوریتم تقسیم با بازیابی در شکل زیر آمده است.

1. $w_0 = x$
2. for $i = 0$ to $n-1$ do
 - 2.1. $w_i = 2w_i - d$
 - 2.2. if $(w_i \geq 0)$
 $w_{i+1} = w_i; q_{n-1-i} = 1;$
 - else
 $w_{i+1} = w_i + d; q_{n-1-i} = 0;$

می‌توان با انجام یک تغییر ساده سرعت این الگوریتم را افزایش داد. ایده اصلی این الگوریتم تقسیم -که Non-Performing نامیده می‌شود- ذخیره نکردن باقیمانده پاره‌ای موقت است تا دیگر در صورت منفی شدن آن نیازی به بازیابی نباشد. بنابراین در هر مرحله مقدار $2w_i$ یا مقدار $2w_i - d$ در رجیستر ذخیره می‌شود. این الگوریتم تقسیم در شکل زیر آمده است.

1. $w_0 = x$
2. for $i = 0$ to $n-1$ do
if $(2w_i - d \geq 0)$
 $w_{i+1} = 2w_i - d; q_{n-1-i} = 1;$
else
 $w_{i+1} = 2w_i; q_{n-1-i} = 0;$

تقسیم بدون بازیابی

الگوریتم تقسیم بدون بازیابی به این صورت است که باقیمانده پاره‌ای منفی که در یک مرحله به دست می‌آید را می‌پذیریم (یعنی بلافاصله با انجام عملیات جمع آن را تصحیح نمی‌کنیم) و تصحیح آن را در مرحله بعد با انجام عملیات جمع انجام می‌دهیم. الگوریتم تقسیم بدون بازیابی در شکل زیر آمده است. از آنجا که باقیمانده نهایی باید حتماً مثبت باشد (توجه کنید که در اینجا تقسیم اعداد مثبت را توضیح می‌دهیم، در نتیجه باقیمانده نهایی باید مثبت باشد)، بنابراین در مرحله آخر اگر باقیمانده پاره‌ای منفی باشد یک مرحله تصحیح نهایی داریم (مرحله 4).

1. $w_0 = x$
2. $w_1 = 2w_0 - d$
3. for $i=1$ to $n-1$ do
 - 3.1. if $(w_i \geq 0)$
 $w_{i+1} = 2w_i - d; q_{n-1-i} = 1;$
else
 $w_{i+1} = 2w_i + d; q_{n-1-i} = 0;$
4. if $(w_n < 0)$
 $w_n = w_n + d; q_0 = 0;$
else
 $q_0 = 1;$

با طراحی مسیره‌داده و واحد کنترل یک تقسیم‌کننده با بازیابی برای تقسیم یک عدد ۱۰ بیتی بر یک عدد ۵ بیتی طراحی کنید و آن را با Verilog مدل‌سازی کنید. با اعمال ورودی‌های کافی عملکرد تقسیم‌کننده‌ی خود را تست کنید.

روش ارزیابی:

- پیاده‌سازی تقسیم‌کننده ۱۰۰ نمره دارد
 - ۲۵ نمره طراحی مسیر داده و واحد کنترل (به همراه کد وریلاگ باید بارگذاری شود)
 - ۱۵ نمره روش کدینگ (مسیر داده به صورت ساختاری و واحد کنترل به روش هافمن)
 - ۴۰ نمره صحت طراحی با بردارهای آزمون توسط شما
 - ۲۰ نمره صحت طراحی با بردارهای آزمون توسط دستیاران آموزشی

- هر گونه کپی با نمره ۰ جریمه خواهد شد