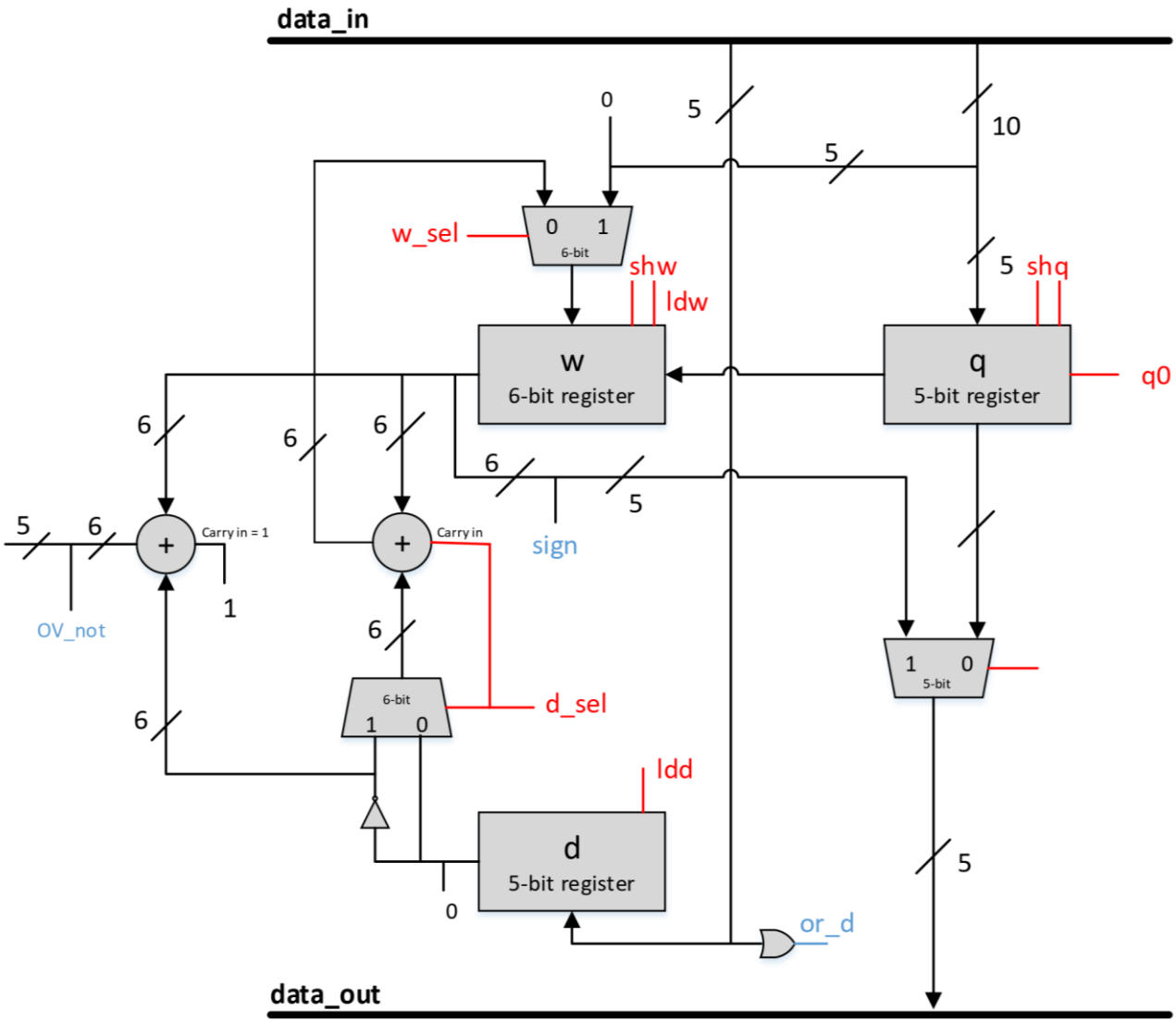
اعضای گروه:
سـوگل گـودرزی ۸۱۰۱۹۸۴۶۷
عرفـان پـنـــاهی ۸۱۰۱۹۸۳۶۹

مسیر داده (Datapath)

برای پیاده سازی یک تقسیم کننده با بازیابی ابتدا مسیر داده مناسب را به صورت زیر تشکیل می دهیم:



❖ **رجیسترها** : این مسیر داده شامل سه رجیستر است. یک رجیستر به نام d برای نگهداری مقسوم علیه (divisor) ، یک رجیستر به نام w برای نگهداری پنج بیت سمت چپ مقسوم قرار داده ایم که در انتها این رجیستر حامل باقیمانده نیز هست. رجیستر دیگری که قرار داده شده، رجیست q است که حاوی ۵ بیت سمت راست مقسوم علیه است. در انتهای کار این تقسیم کننده، رجیستر q حامل خارج قسمت تقسیم خواهد بود.

d_reg : برای ذخیره مقسوم علیه (فقط عمل load در این رجیستر انجام می شود).

w_reg : برای ذخیره ۵ بیت پرارزش مقسوم و یک صفر در سمت چپ آن (عملیات shift و load انجام می شود). سریال ورودی آن از بیت سمت چپ رجیستر q گرفته شده

q_reg : برای ذخیره ۵ بیت سمت راست مقسوم (عملیات shift و load انجام می شود). سریال ورودی آن همان q0 است.

❖ **مالتیپلکسرها :** در ادامه برای مسیر داده، سه مالتیپلکسر قرار داده ایم. مالتیپلکسری که ورودی های آن از رجیستر d گرفته شده است در واقع برای انتخاب کردن عملیت جمع یا تفریق قرار داده شده. خروجی این مالتیپلکسر به یک جمع کننده ۶ بیتی متصل شده است. کری ورودی و سیگنال سلکتور مالتیپلکسر هر دو به هم متصلند. همانطور که از مدار نیز مشخص است می بینیم که اگر d_sel مقدار صفر داشته باشد بیت های d با کری ورودی صفر وارد جمع کننده می شوند و عملیات جمع انجام می شود. حال اگر d_sel مقدار ۱ را داشته باشد آن گاه بیت های رجیستر d معکوس خواهند شد و کری ورودی در ادر نیز ۱ است پس عملکردی که در ادر انجام می شود این است که مقدار w با 2's complement مقدار رجیستر d جمع می شود که بیانگر همان عمل تفریق است. مالتیپلکسر دیگری که قرار داده ایم برای این است که در ابتدای شروع کار مدار، مقدار ورودی را از باس بگیریم و در رجیستر w قرار دهیم و در سیکل های بعدی، حاصل جمع یا تفریق را در w قرار دهیم. مالتیپلکسر دیگری که قرار داده ایم برای این است که خروجی ها را به صورت ۵ بیت ۵ بیت در دو کلاک متوالی روی باس خروجی قرار دهیم.

mux1 : برای مشخص کردن عملیت جمع یا تفریق با مقسوم علیه

mux2 : برای تعیین گرفتن ورودی رجیستر w در شروع عملکرد مدار و لود کردن حاصل جمع یا تفریق w و d در سیکل های بعدی مدار

mux3 : برای قرار دادن خروجی ها روی باس خروجی در دو کلاک متوالی

❖ **جمع کننده ها :** در مسیر داده مدنظر یک جمع کننده برای اینکه عمل جمع یا تفریق انجام شود قرار داده ایم. جمع کننده دیگر به منظور بررسی رخ دادن حالت سرریز یا overflow قرار داده شده است. در این حالت از سیگنالی که خروجی رجیستر d را NOT کرده ایم، یکی از ورودی های ادر را تامین می کنیم. ورودی دیگر را نیز از همان رجیستر w قرار می دهیم. کری ورودی این ادر را ۱ قرار داده ایم چون قرار است حاصل تفریق ۵ بیت پرارزش مقسوم و مقسوم علیه را بررسی کنیم. دقت شود که برای این منظور از ساین اکستند استفاده شده. حال برای بررسی حالت سرریز باید MSB را بررسی کنیم. اگر این بیت مقدار صفر داشته باشد یعنی (۵ بیت پرارزش مقسوم علیه $\geq$ ۵ بیت پرارزش مقسوم) پس سرریز رخ می دهد و اگر MSB مقدار ۱ داشته باشد آن گاه (۵ بیت پرارزش مقسوم علیه $\leq$ ۵ بیت پرارزش مقسوم) است و سرریز نخواهیم داشت.

جمع کننده اول : برای انجام عملیات جمع یا تفریق w و q

جمع کننده دوم : برای تشخیص وقوع حالت سرریز یا overflow . کری ورودی این ادر را ۱ قرار داده ایم چون قرار است حاصل تفریق ۵ بیت پرارزش مقسوم و مقسوم علیه را بررسی کنیم. دقت شود که برای این منظور از ساین اکستند استفاده

شده. حال برای بررسی حالت سرریز باید MSB را بررسی کنیم. اگر این بیت مقدار صفر داشته باشد یعنی (۵ بیت پرارزش مقسوم علیه ≥ ۵ بیت پرارزش مقسوم) پس سرریز رخ می دهد و اگر MSB مقدار ۱ داشته باشد آن گاه (۵ بیت پرارزش مقسوم علیه ≤ ۵ بیت پرارزش مقسوم) است و سرریز نخواهیم داشت.

❖ **گیت OR** : این گیت در واقع بیت های مقسوم علیه را قبل از اینکه وارد رجیستر d شوند با هم or می کند تا بررسی کند که آیا مقسوم علیه صفر است یا خیر.

سیگنال هایی که از مسیر داده به کنترلر داده می شود :

sign : این سیگنال در واقع همان بیت سمت چپ w است که برای بررسی اینکه عدد حاصله منفی است یا مثبت، به کنترلر برای کنترل کردن سیگنال های دیگر داده می شود. sign=w[5]

d_or : این سیگنال برای بررسی این است که مقسوم علیه صفر است یا خیر در واقع برای بررسی اینکه حالت divide by zero داریم یا نه، از این سیگنال در کنترلر استفاده می کنیم.

ov_not : این سیگنال برای خروجی ادری که برای بررسی سرریز قرار دادیم است. اگر این سیگنال ۱ باشد در واقع سرریز رخ نمی دهد و بالعکس.

سیگنال هایی که از کنترلر به مسیر داده وارد می شوند :

سیگنال های load برای رجیسترها : ldd , ldw , ldq

سیگنال های select برای مالتیپلکسرها : d_sel , w_sel , out_sel

سیگنال های شیفت برای رجیسترها : shw , shq

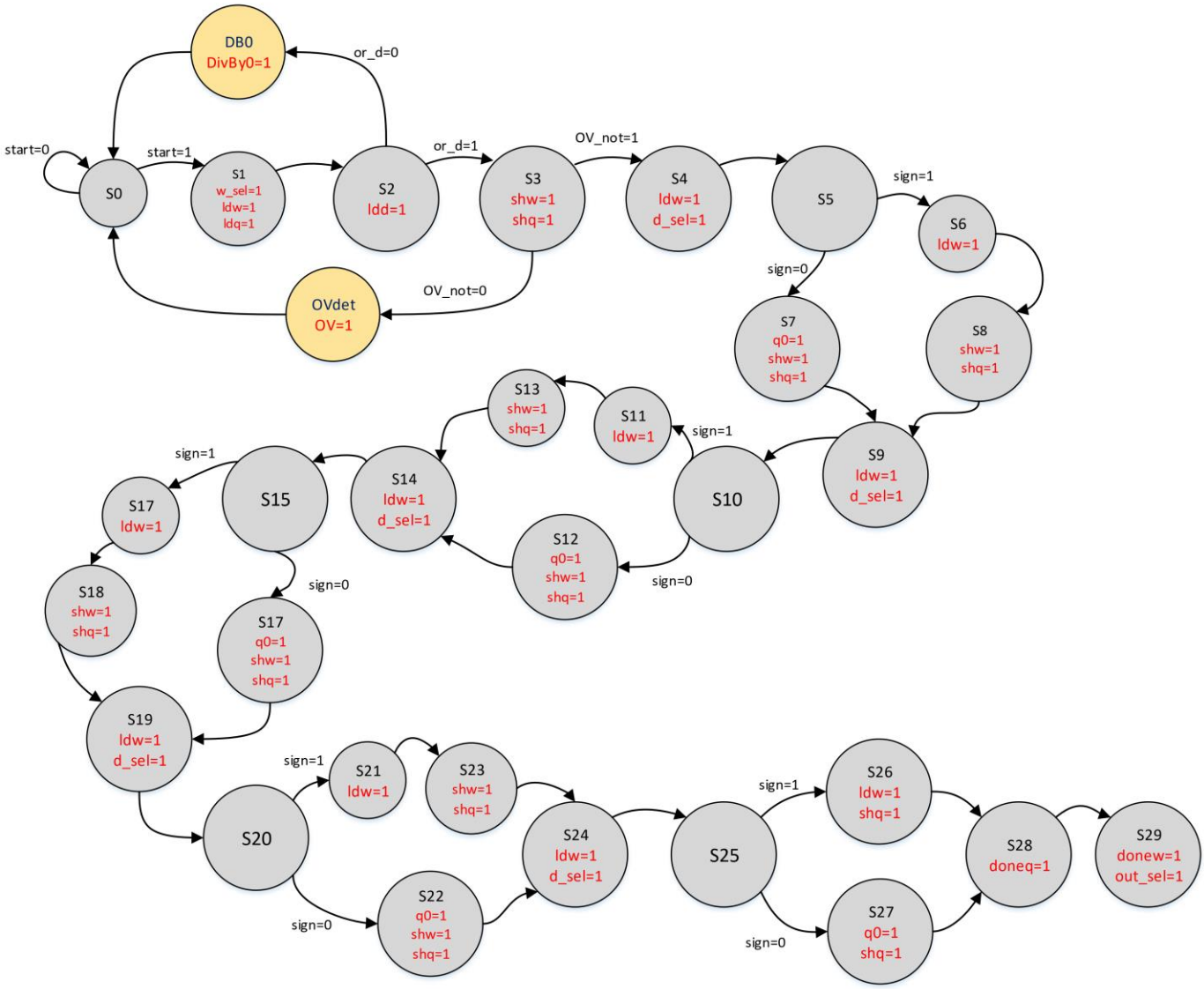
q0 : این سیگنال برای تعیین کردن این است که اگر حاصل تفریق w و d مثبت بود، بیت q[0] را ۱ کنیم.

کنترلر (Controller)

برای این قسمت داده ها را از باس ورودی به این صورت گرفته ایم که ابتدا ۱۰ بیت مقسوم را گرفته و ۵ بیت راست آن را در q و ۵ بیت چپ آن را با یک بیت صفر در سمت چپ آن، به w می دهیم. در کلاک بعدی، مقسوم را از باس ورودی می گیریم و در d قرار می دهیم. برای قرار دادن خروجی نیز به این صورت عمل کردیم که با دو کلاک متوالی خروجی هارا روی باس گذاشته ایم. در کلاک اول، خارج قسمت را روی باس قرار می دهیم و می توانیم با یک شدن doneq تشخیص دهیم که خارج قسمت روی باس خروجی قرار گرفته است. در کلاک بعدی باقیمانده را قرار می دهیم و donew را یک می کنیم تا بدانیم باقیمانده در باس خروجی قرار دارد.

روند کلی استیت دیاگرام زیر به این گونه است که بعد از لود کردن مقادیر رجیسترها و بررسی حالت divide by zero و overflow ، عملیات مورد نظر (شیفت دادن و جمه یا تفریق کردن و یک کردن q0 در صورت نیاز) را ۵ بار انجام می دهد و سپس خروجی های نهایی بر روی باس قرار می گیرند.

State Diagram :



در شکل زیر نیز ارتباط بین قسمت مسیر داده و کنترلر مشخص شده است :