

Practical 3



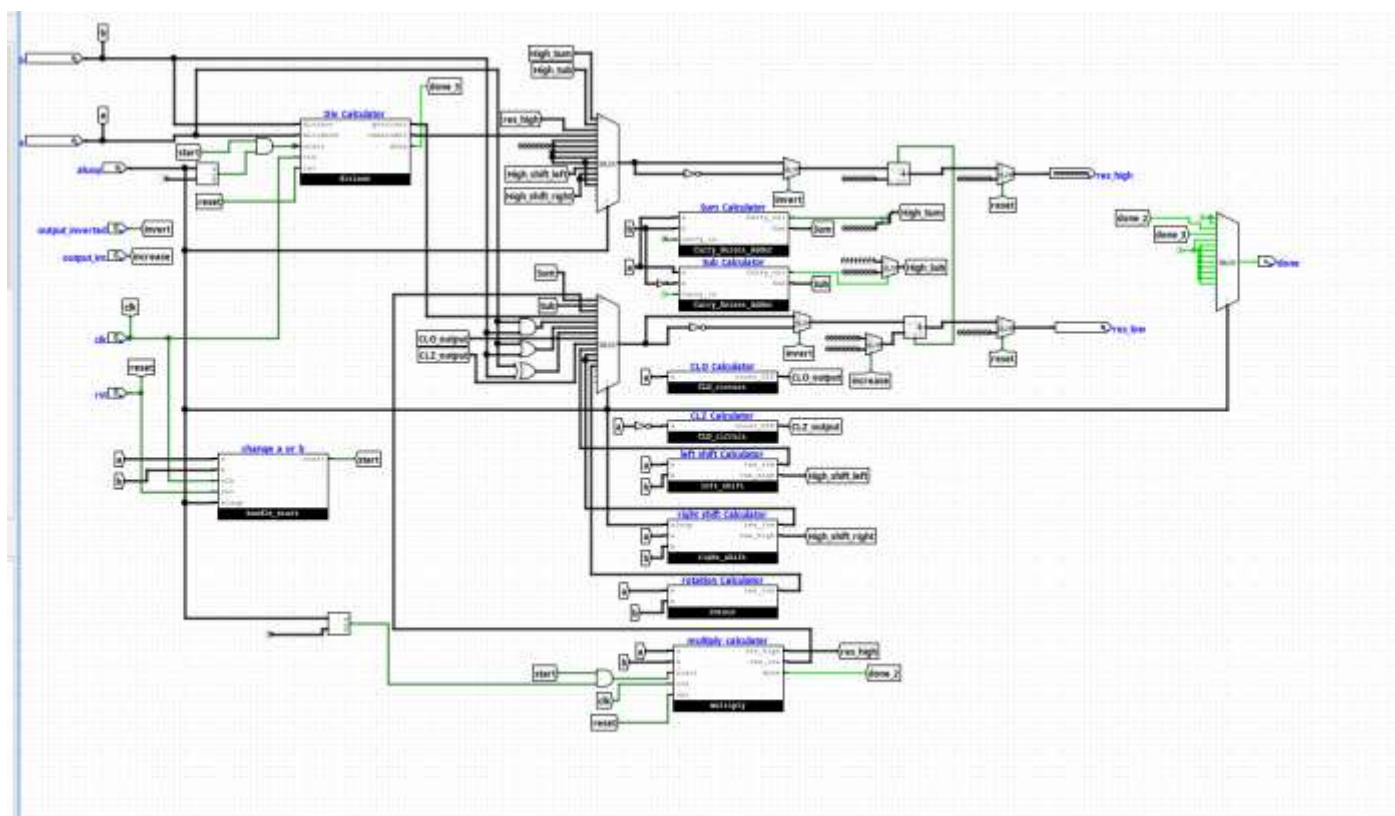
گزارش تمرین عملی :

در این تمرین به طراحی یک ALU پرداختیم. عملیات های لازم در قسمت های جداگانه طراحی شدند. در قسمت اصلی یا main با استفاده از یک MUX که سیگنال select آن به aluop متصل است اینکه خروجی کدام قسمت به عنوان جواب در نظر گرفته شود بررسی می شود. همچنین برای هندل کردن سیگنال output_inverted بین خروجی MUX اصلی و not این خروجی با استفاده از MUX با سیگنال select = inverted انتخاب می کنیم و بین خروجی آن و یکی بیشتر از آن (با adder ایجاد شده) با MUX با سیگنال select = output_inc انتخاب می کنیم. به این نکته نیز توجه داریم که اگر عملیات inc در ۳۲ بیت پایین carry داد این carry را به ۳۲ بیت بالاتر منتقل کنیم.

برای اینکه در عملیات های ضرب و تقسیم نیز در یک کلاک start=1 شود و بعد start = 0 قسمت handle_start طراحی کردیم که با رصد ورودی های a,b هر گاه که این ورودی ها تغییر کردند مقدار start برابر یک شده و اگر تغییری رخ نداد صفر می شود.

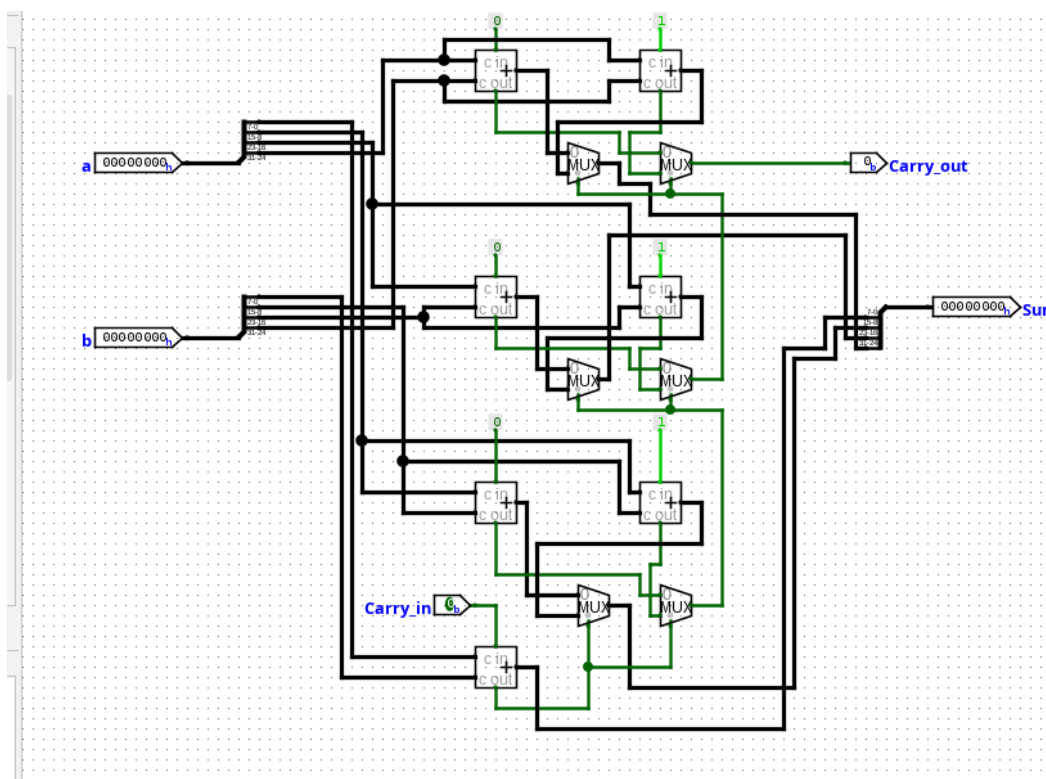
همچنین عملیات های ضرب و تقسیم با پایان یافتنشان یک سیگنال done خروجی می دهند و با استفاده از MUX با توجه به aluop سیگنال done مدار اصلی بر اساس آنها تعیین می شود. (البته توجه داریم عملیات هایی که یک کلاکه هستند سیگنال done آنها به ۱ متصل است)

در زیر نمای کلی مدار اصلی را مشاهده می کنیم:



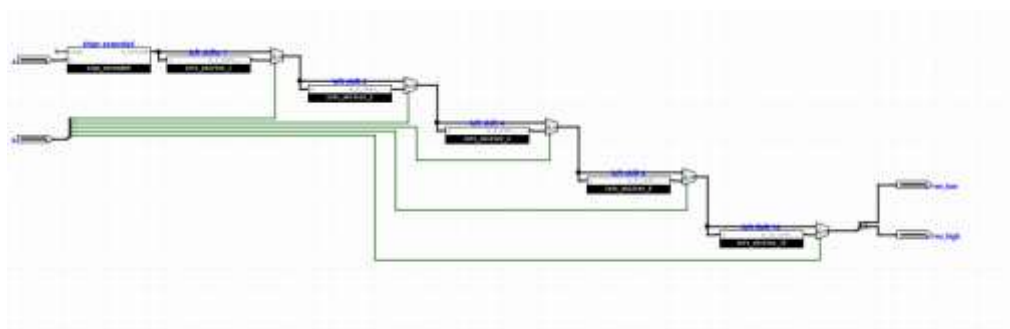
حال به توضیح بخش های درونی مدار می پردازیم:

عملیات جمع با استفاده از carry_select_adder ای که دارای ۴ قسمت ۸ بیتی است و با استفاده از MUX بین عدد با carry=1, carry=0 انتخاب می کنیم طراحی شده است. برای تفریق نیز عدد دوم را not کرده و با دادن cin=1 در اصل مکمل دوم عدد دوم را با عدد اول جمع می کنیم و همچنین برای ۳۲ بیت بالاتر با توجه به borrow خروجی داده شده عمل می کنیم.



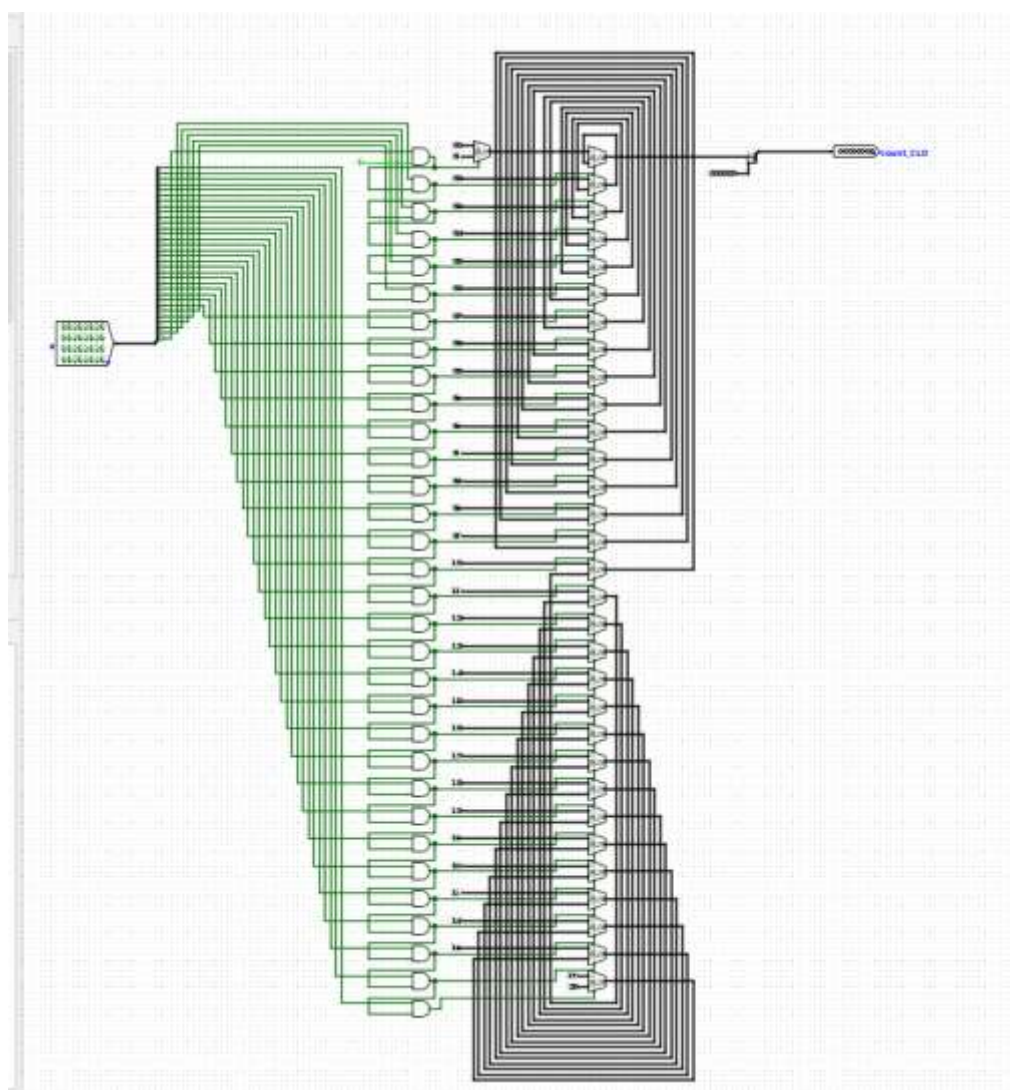
عملیات های منطقی or, and, xor نیز با گیت های خود لاجیسیم طراحی شدند.

برای انجام انواع شیفت ها و rotate از یک ساختار مشخص استفاده کردیم یعنی برای هر کدام شیفت و rotate ۱ بیتی و ۲ بیتی و ۴ بیتی و ۸ بیتی و ۱۶ بیتی را با توجه به تعریف عملیات ساختیم سپس با ۵ لایه MUX با بیت select بیت های صفر تا ۴ عدد b (که با splitter جدا شده است) به اندازه لازم عملیات ها انجام می شود.



برای انجام عملیات CLO ابتدا هر دو بیت متوالی را با یکدیگر and کردیم تا یک بودن دوتای آنها مشخص شود سپس با استفاده از MUX هایی که بیت select آنها خروجی گیت های and است مقدار مناسب را خروجی می دهیم. توجه داریم جواب نهایی از اولین MUX به دست می آید یعنی اگر دو بیت اول یک بودند MUX اول از خروجی MUX بعدی جواب را می گیرد تا الی آخر.

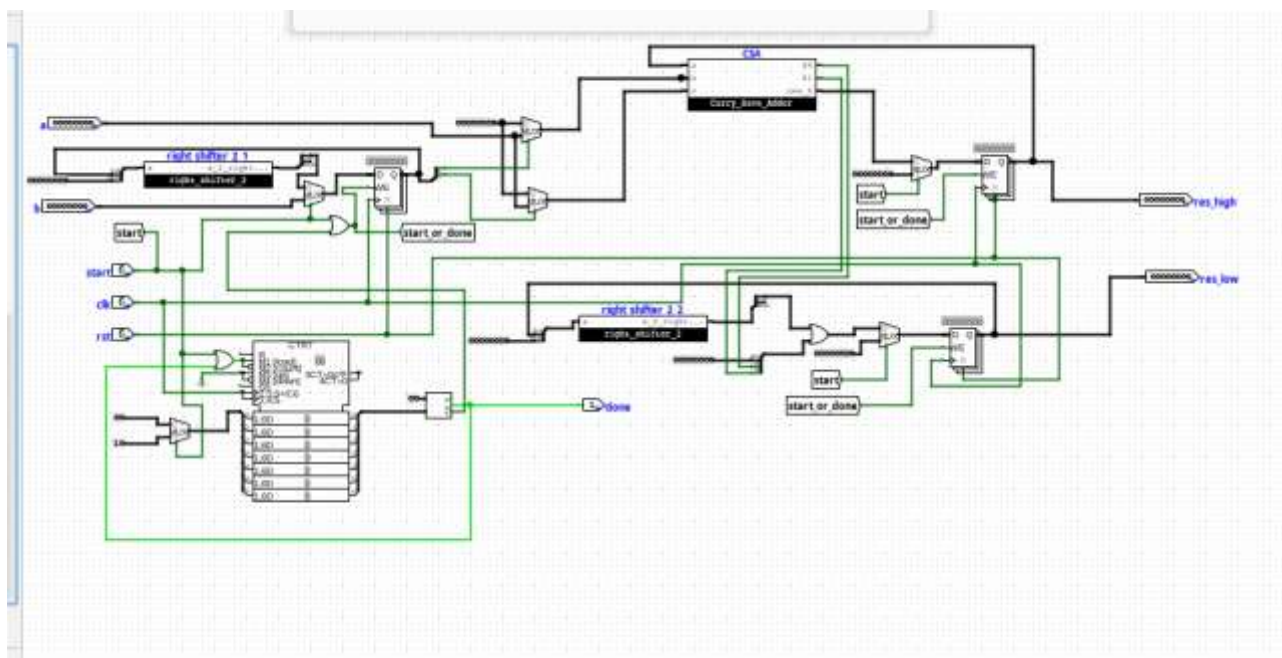
برای انجام عملیات CLZ نیز عدد را نات کرده و از همان CLO استفاده می کنیم تا مقدار صفر های پشت سر هم از راست را به ما بدهد.



برای انجام عملیات ضرب در ۱۶ کلاک با استفاده از carry_select_adder ای که طراحی کردیم یک carry_saved_adder ۳۲ بیتی ساختیم که همزمان ۳ عدد ۳۲ بیتی را با یکدیگر جمع می کند. حال در هر کلاک ضرب



شونده را یکبار یکی و یکبار دوتا نسبت به قبل و البته با توجه به بیت حال حاضر ضرب شونده شیفت داده و با جواب محاسبه شده تا کنون جمع می‌کنیم. در هر کلاک ۲ بیت تثبیت می‌شود بنابراین بعد از ۱۶ کلاک ۳۲ بیت پایینی تثبیت شده و ۳۲ بیت بالایی نیز خروجی CSA است. توجه داریم که با شمارنده ۱۶ کلاکه شدن عملیات را کنترل می‌کنیم.



برای انجام عملیات تقسیم نیز از همان تقسیم کننده ی تمرین دوم عملی بهره بردیم.

در زیر پاسخی که script تست به ما داده است را مشاهده می‌کنیم:

```
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating HDL file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating HDL file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] WARN com.cburch.logisim.fpga.gui.Reporter - Component "Counter" in circuit "erfan.circ" is not a standard component.
main] WARN com.cburch.logisim.fpga.gui.Reporter - Component "Register" in circuit "erfan.circ" is not a standard component.
main] WARN com.cburch.logisim.fpga.gui.Reporter - Component "Register" in circuit "erfan.circ" is not a standard component.
main] WARN com.cburch.logisim.fpga.gui.Reporter - Component "Register" in circuit "erfan.circ" is not a standard component.
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating HDL file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating HDL file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating HDL file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating script file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating script file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating script file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
main] INFO com.cburch.logisim.fpga.gui.Reporter - Creating script file: /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp

W3/tb.v /home/mohamamdreza/logisim_evolution_workspace/erfan.circ.tmp
CCEPTEED
500 / 500
mohamamdreza@nylove:~/CA/SUT_CA_4032_ProfAsadi_Judgement_System-master$
```