توضیح عملکرد:

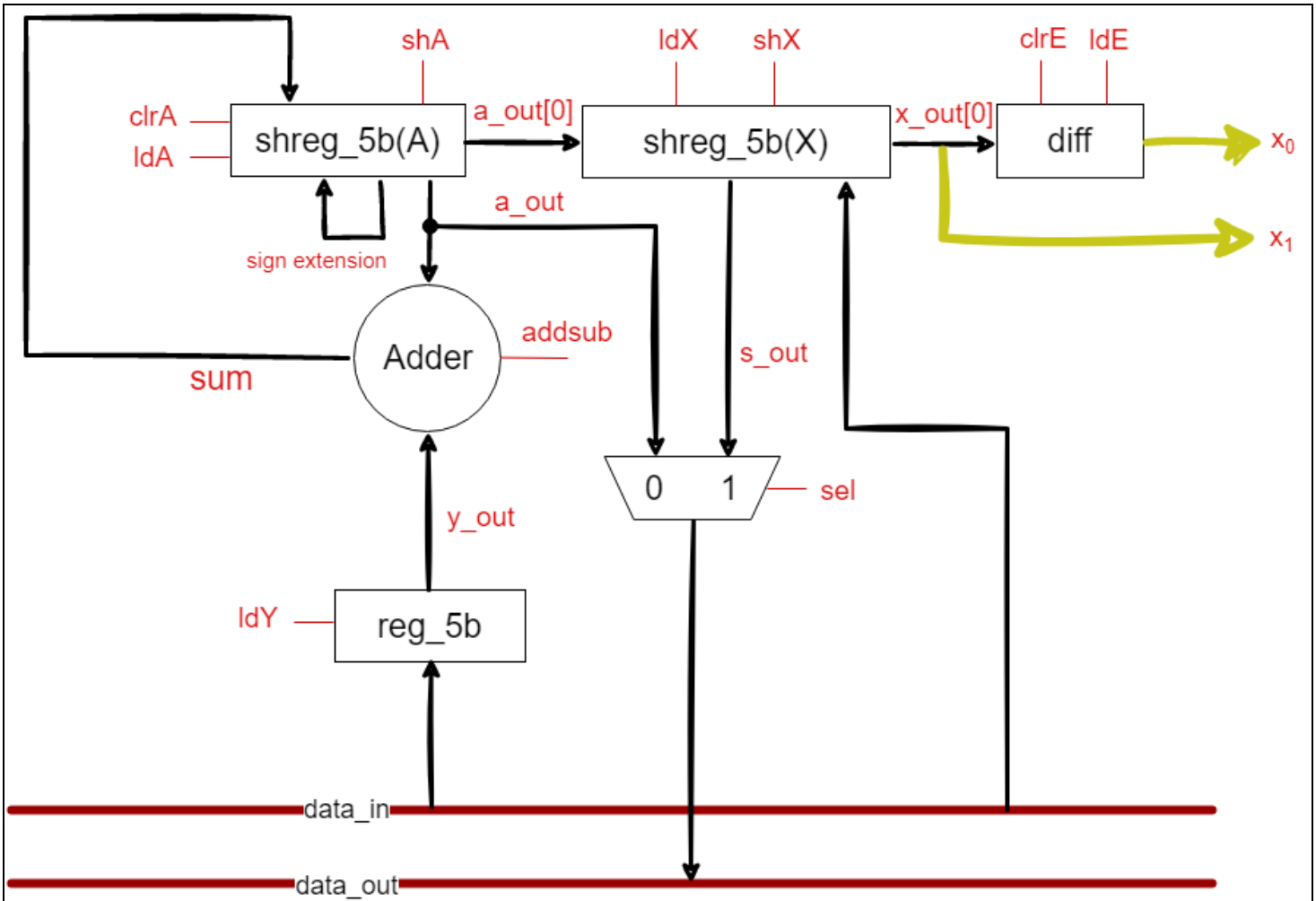
ورودی ها را روی data_in قرار می دهیم ورودی اول وارد multiplicand(reg 5_bit Y) و ورودی دوم وارد multiplier(shreg 5_bit X) قرار می دهیم.

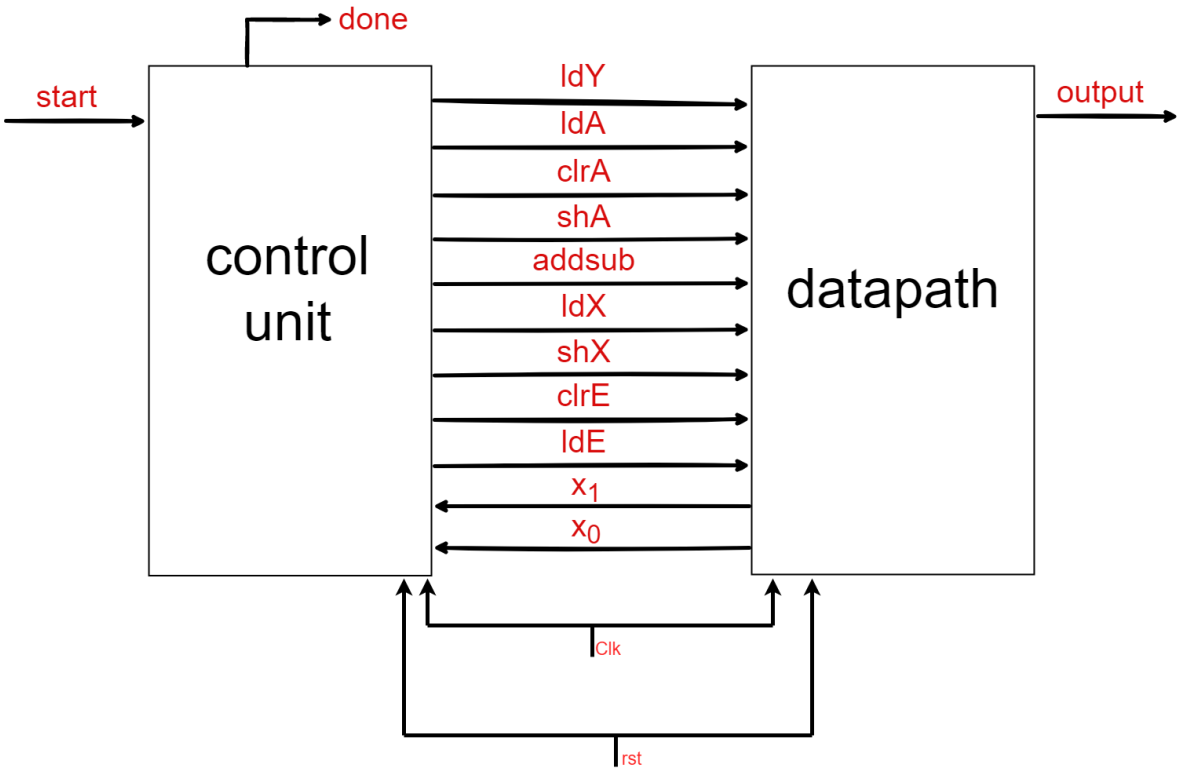
و بعد از آن A و x1 را رست می کنیم بعد میایم و شروع می کنیم و بیت های x1 و x0 را بررسی می کنیم و بعد در A می نویسیم.

- اگر یکسان بودن هیچ کاری نمی کنیم
- اگر 01 بودن جمع می کنیم
- اگر 10 بودن تفریق می کنیم

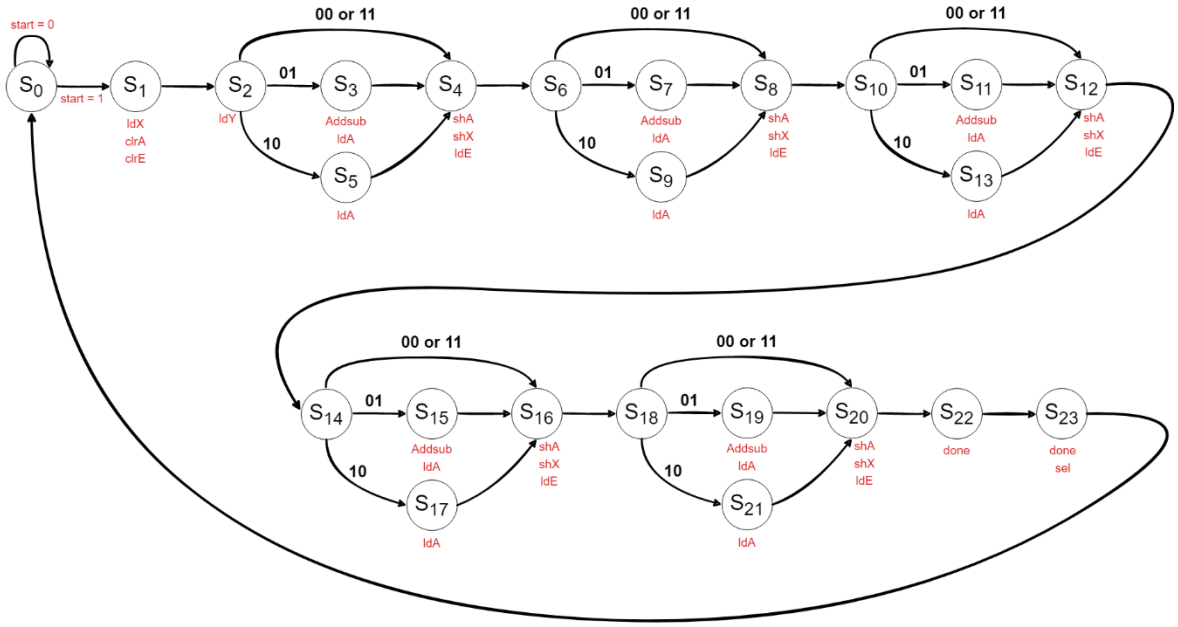
بعد برای جمع اعداد برا اینکه اعداد متناظر عدد خود جمع شن یه شیفت به راست هم می دهیم.

و بعد از اینکه کار تموم شد ابتدا ۵ بیت پر ارزش و سپس ۵ بیت کم ارزش در خروجی data_out نشان داده می شود آن هم وقتی مقدار done برابر یک هستش.


shA
ldX
shX
clrE
ldE
clrA
ldA
shreg_5b(A)
a_out[0]
shreg_5b(X)
x_out[0]
diff
x_0
x_1
a_out
sign extension
Adder
addsub
sum
s_out
0
1
sel
y_out
ldY
reg_5b
data_in
data_out


done
start
control unit
ldY
ldA
clrA
shA
addsub
ldX
shX
clrE
ldE
x_1
x_0
datapath
output
Clk
rst

State Machine:



ماژول های استفاده شده:

```
adder add_4b (a_out, y_out, 1'b0, addorsub , cout, sum);

reg_5b Y(data_in, ldY, clk, y_out);

dff E(x1 , clrE, ldE, clk, x0);

shreg_5b A(sum, a_out[4], clrA, ldA, shA, clk, a_out);

shreg_5b X(data_in, a_out[0], 1'b0, ldX, shX, clk, x_out);

mux_2_to_1 mux(a_out, x_out, sel , data_out);
```

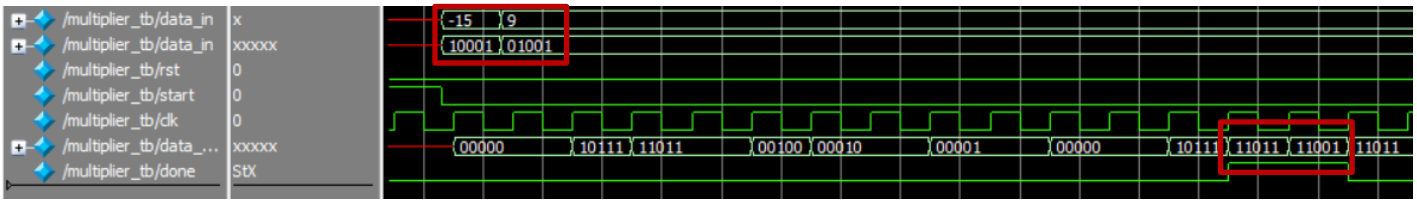
- یه ماژول جمع یا تفریق ()
- رجیستر ۵ بیتی
- رجیستر یک بیتی
- دو تا شیفت رجیستر ۵ بیتی
- مولتی پلکسر دو به یک

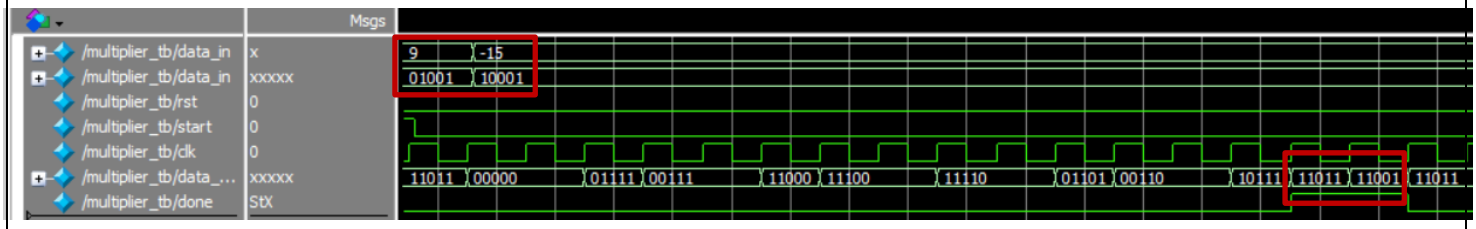
تست ها:

1)

10001 * 01001 =1101111001

-15 *9 = -135

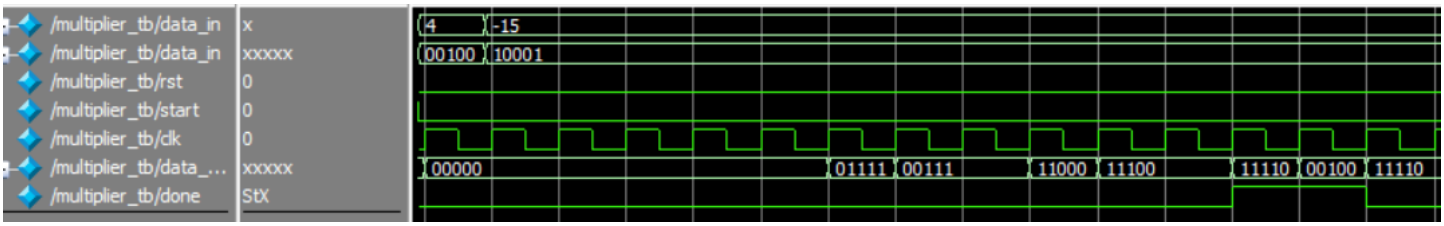




در اینجا چک کردیم که اگه ترتیب ورودی ها اگر فرق کنه هم باز جواب یکیه.

2) 10001 * 00100 =1111000100

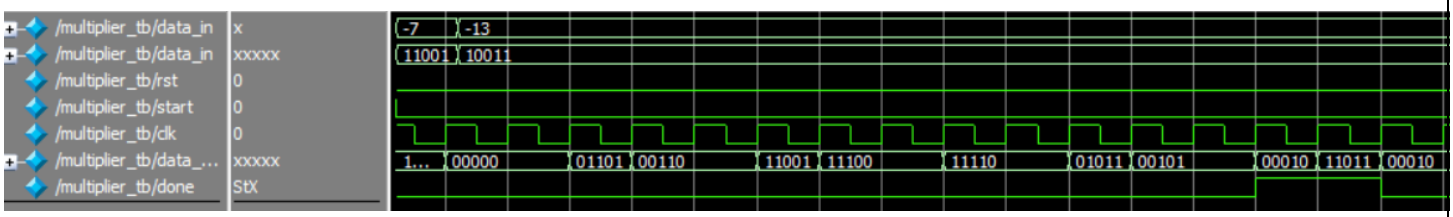
-15*4=-60



3)

11001 * 10001 = 0001011011

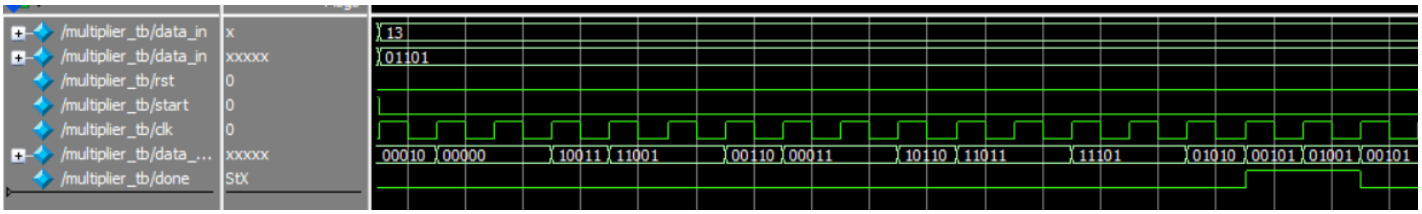
-7 * -13 = 91



4)

13 * 13 = 169

01101 * 01101 = 0010101001



5)

-11 * -14 = 151

10101 * 10010 = 0010010111