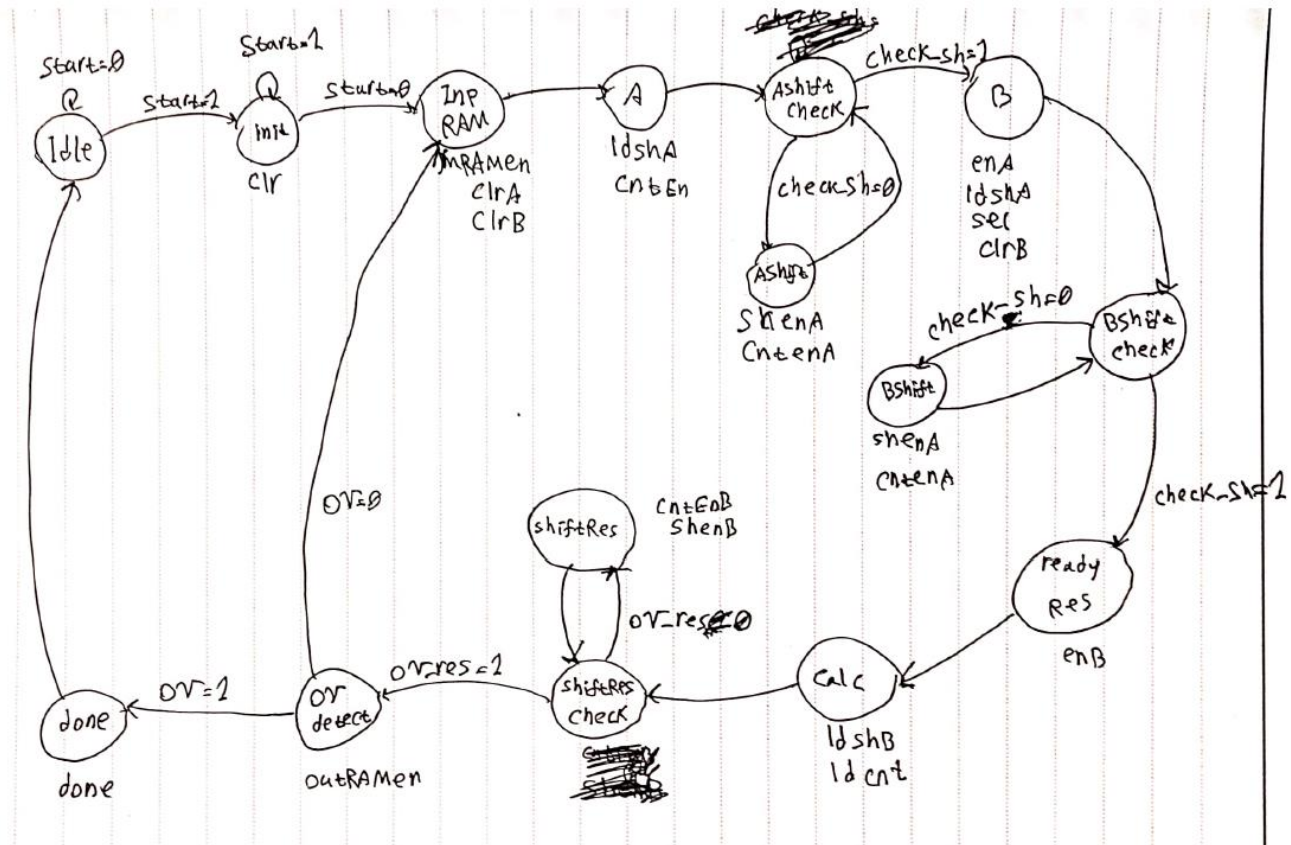
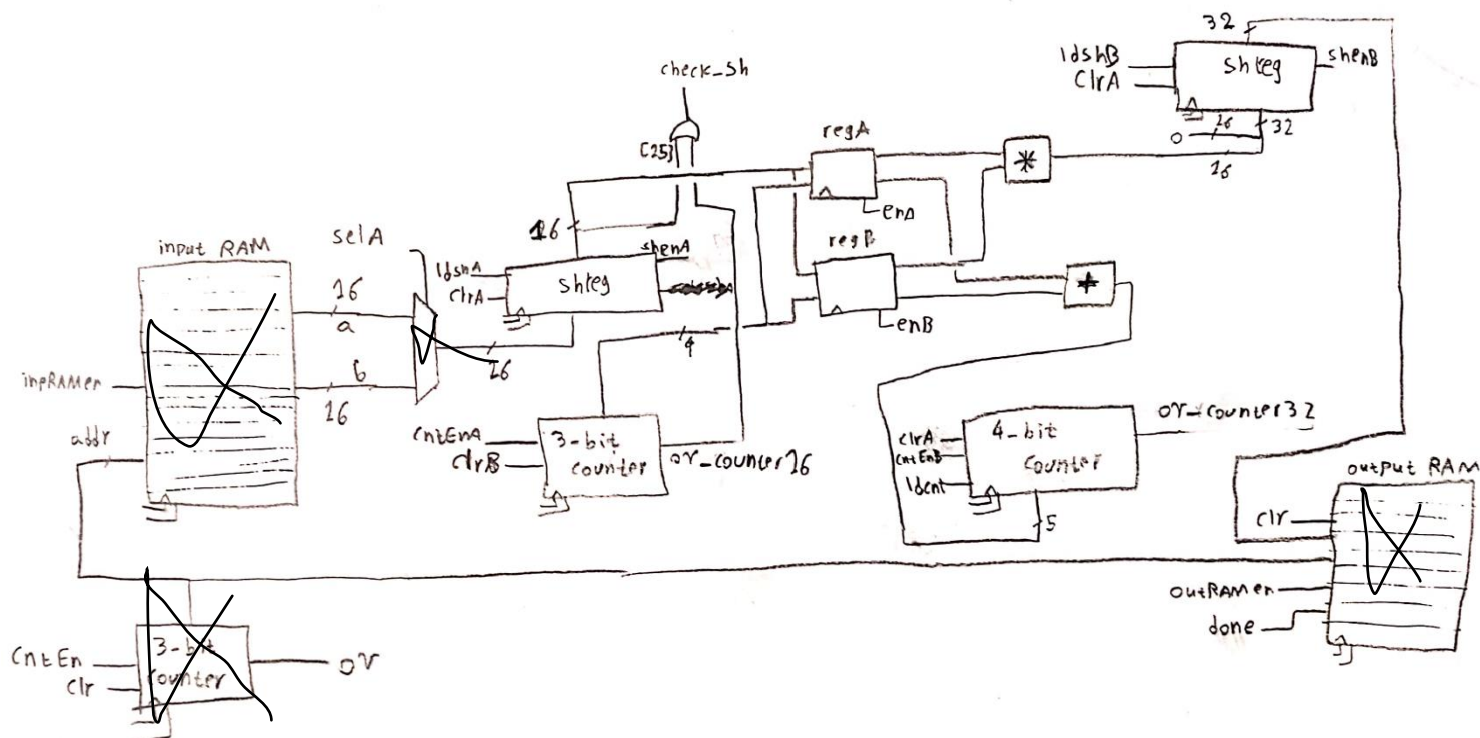


سپهر جمالی ۱۴۰۰/۱۰/۱۱

محمد مهدی صمدی ۱۴۶۵/۱۰/۱۱



کنترلر به صورت moore طراحی شده. در استیت های idle و init پالس استارت دیده میشود. در استیت های InpRam و ovDetect به ترتیب خواندن و نوشتن رم ها صورت میگیرد. از استیت A تا shiftResCheck محاسبات مربوط به ضرب انجام میشود به طوریکه عدد اول از مموری پردازش شده و سپس به سراغ عدد دوم می رویم و سپس به محاسبه خروجی میپردازیم. سه جفت استیت مربوط به شیفت مربوط به shift register و counter مربوط به آنها میباشد. در نهایت در استیت done نوشتن در مموری خروجی تکمیل شده.



دو عدد a و b را از مموری ورودی میخوانیم. ابتدا برای a و بعد از آن برای b بیت های مفید را به کمک shift register بدست میاوریم و با counter تعداد صفر ها را ذخیره میکنیم. در واقع counter زمانی متوقف میشود که اولین ۱ را ببینیم یا نهایتا تا هشت بیت صفر بشمریم که در این حالت خروجی بدون خطا تولید میشود. برای خروجی نهایی از shift register ۳۲ بیتی استفاده میکنیم. به طوریکه شونزده بیت مفید به همراه ۱۶ بیت صفر را در سمت چپ قرار میدهیم و در shift register لود میکنیم. سپس به ازای تعداد صفری که برای رسیدن به چپ ترین ۱ اعداد a و b بدست میاوریم در counter لود میکنیم و تا موقعی که به OV نرسیده عدد نهایی را شیفٹ چپ میکنیم. در واقع با این روش به ازای مکمل تعداد صفری که در چپ اعداد قرار داشت در راست عدد نهایی قرار میدهیم. در نهایت خروجی را در مموری خروجی مینویسیم.

data_input.txt

1010011010101001

0010010001000110

1101000111010111

1100010111101010

0110110000011111

1110110010111000

0001010011000001

1000011111010100

0011000011001011

0110111101011110

1001000010010010

1111111100011010

0101101101101010

1101110111000010

1111101000010000

1100001101010111

data_output.txt

17818000

a0d50000

63900000

0af14000

15234000

8f700000

4e8f0000

be6e0000

data_inputA.txt (py file)	data_outputA.txt
1111000101000101	08a63000
0000100100111111	128fc000
0001010011000110	16030000
1110010101010101	0f99c000
1011100000001010	3db98000
0001111010100010	0056d600
0110101011111111	18c3c000
0010010110000111	017a7000
1101110100110111	
0100011111100000	
0000111010100100	
0000010111110110	
0011010110010101	
0111011010111000	
1000110000011110	
0000001010110111	