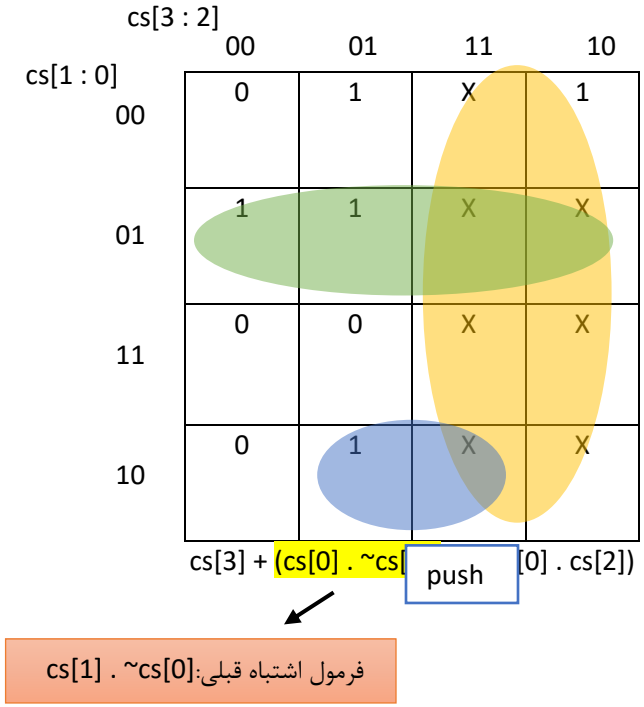
گزارش تمرین دوم درس طراحی کامپیوتری سیستم‌های دیجیتال

معین کرمی: ۸۱۰۱۹۸۵۴۰ سید محمد امین اطیابی: ۸۱۰۱۹۸۵۵۹

مشکلات طراحی در تمرین اول (معین کرمی) :

تنها مشکل در تمرین اول مربوط به خروجی یکی از کارنومپ‌ها که سیگنال push را تولید می‌کرد بود که فرمول درست آن را در زیر مشاهده می‌کنید(قسمت هایلایت شده قسمتی است که تصحیح شده)

روش تصحیح: فقط کافی بود فرمول را تصحیح کنیم.



تست بنچ:

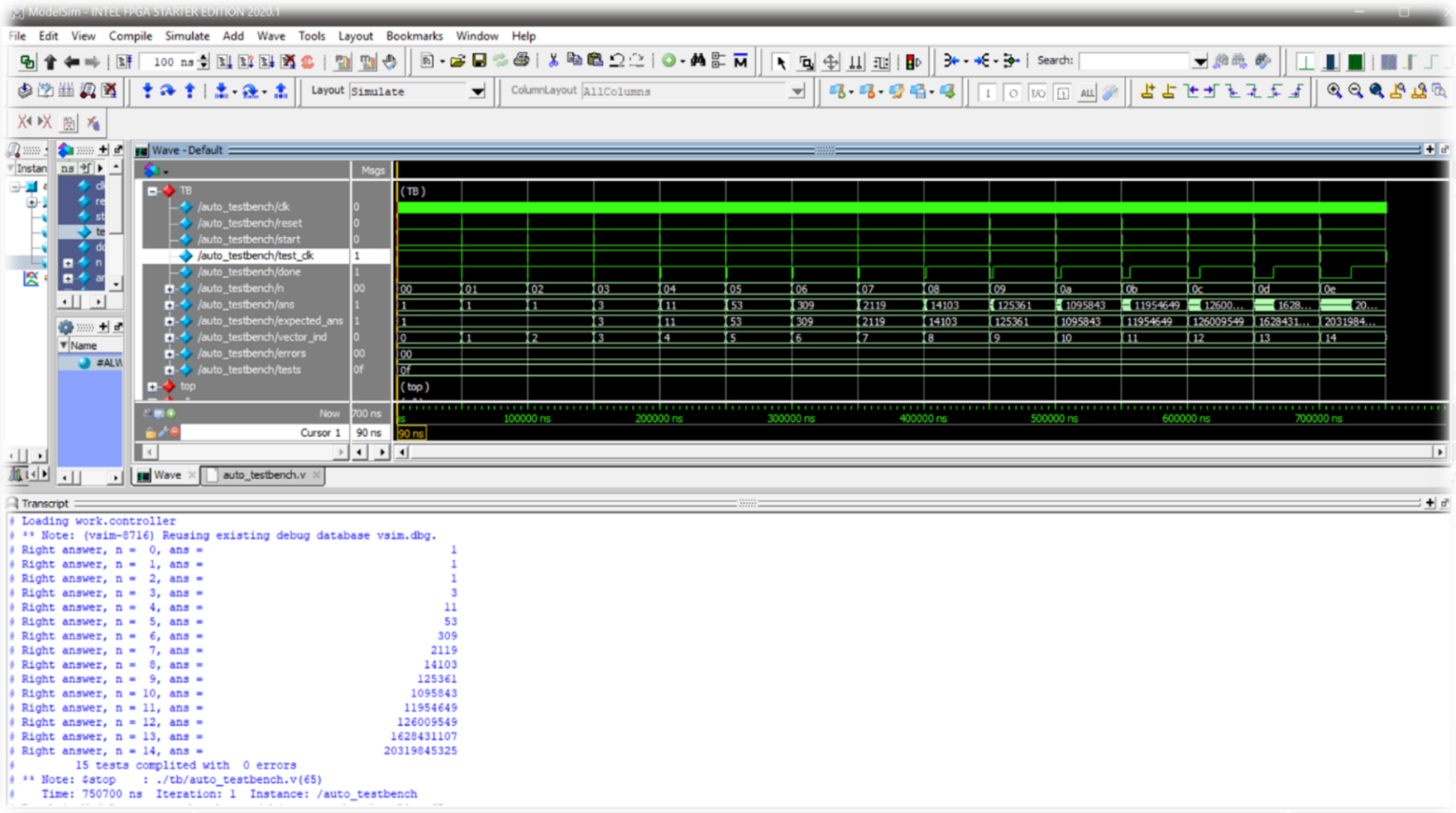
برای تست درستی طراحی از یک تست بنچ self checking استفاده شده است و برای تولید ورودی و خروجی‌های درست متناظر با این ورودی‌ها از یک کد به زبان cpp استفاده شده است که می‌توانید آن را داخل پوشه‌ی sim\model ببینید و همچنین فایل ورودی و خروجی را می‌توانید داخل پوشه‌ی sim\file مشاهده کنید.(تطابق خروجی مدار و خروجی درست و مورد انتظار به طور خودکار داخل تست‌بنچ انجام می‌شود.)

خروجی مدار به ازای ورودی از ۰ تا ۱۴ چک شده است.(طراحی جواب گوی اعداد بزرگتر هم هست ولی اعداد بزرگتر زمان زیادی برای شبیه سازی می‌خواهند و خروجی آن ها ممکن است از 2^{64} بیشتر شود در نتیجه تولید آن‌ها با زبان cpp دشوار خواهد بود.)

نتایج شبیه سازی را می‌توانید در صفحه بعد مشاهده کنید.

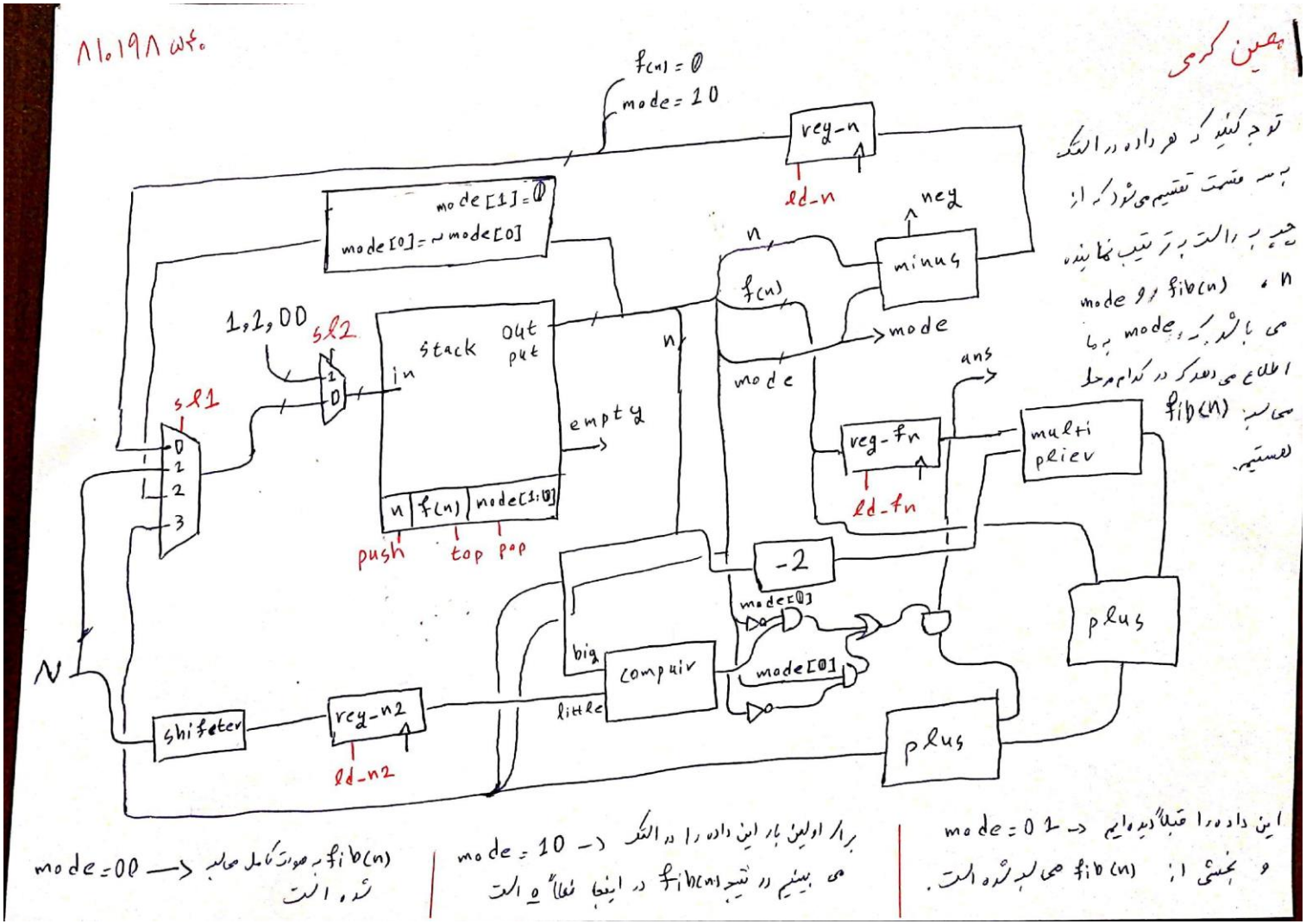
پ.ن: توجه کنید که کارنومپ متناظر سیگنال done کشیده نشده است و دلیل آن این است که این سیگنال فقط در استیت شماره ۰ روشن می‌شود پس داریم:

done = ~cs[0] & ~cs[1] & ~cs[2] & ~cs[3]

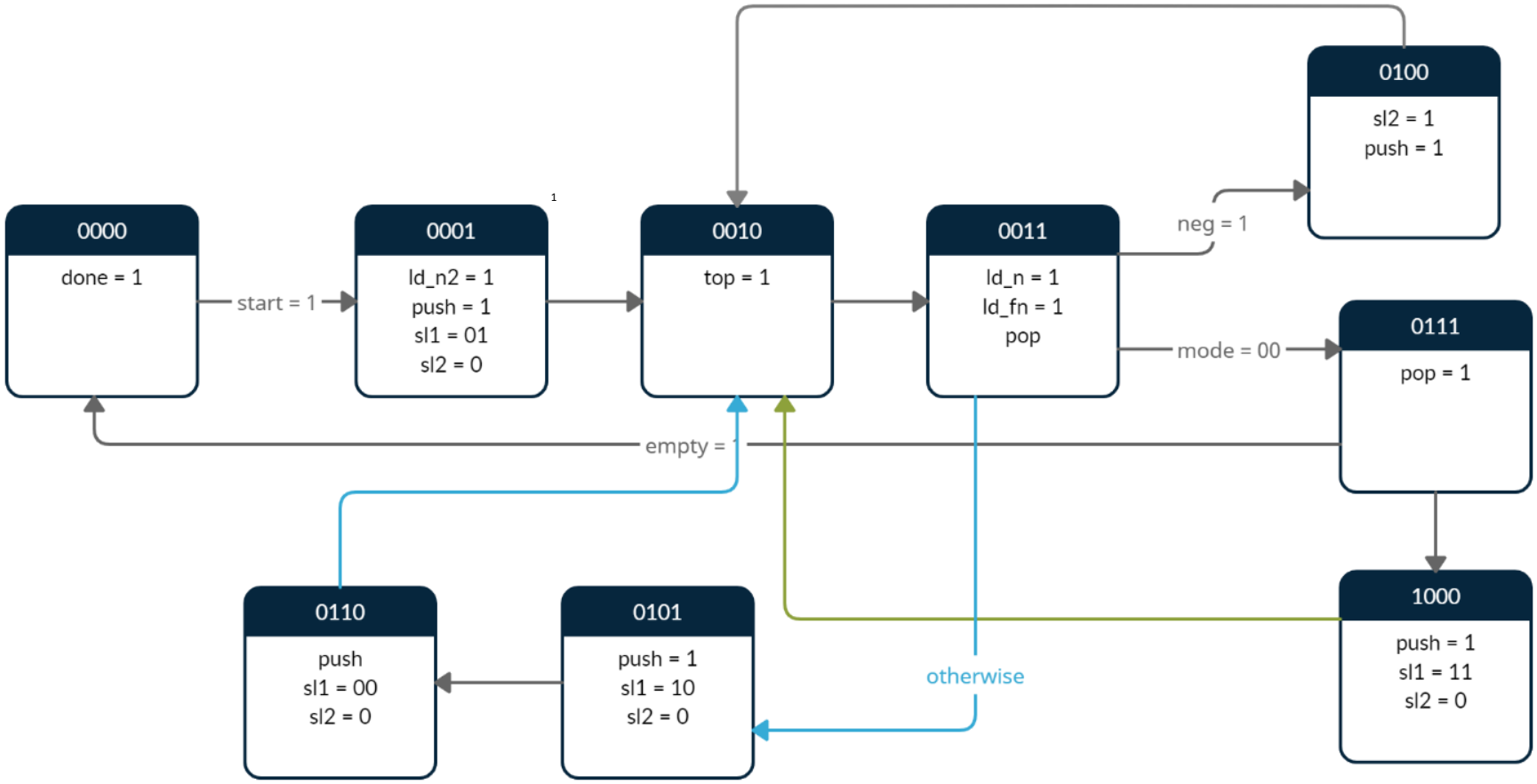

ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default
Msgs
TB
/auto_testbench/clk
/auto_testbench/reset
/auto_testbench/start
/auto_testbench/test_clk
/auto_testbench/done
/auto_testbench/n
/auto_testbench/ans
/auto_testbench/expected_ans
/auto_testbench/vector_ind
/auto_testbench/errors
/auto_testbench/tests
top
Now
Cursor 1
Wave
auto_testbench.v
Transcript
Loading work.controller
** Note: (vsim-8716) Reusing existing debug database vsim.dbg.
Right answer, n = 0, ans = 1
Right answer, n = 1, ans = 1
Right answer, n = 2, ans = 1
Right answer, n = 3, ans = 3
Right answer, n = 4, ans = 11
Right answer, n = 5, ans = 53
Right answer, n = 6, ans = 309
Right answer, n = 7, ans = 2119
Right answer, n = 8, ans = 14103
Right answer, n = 9, ans = 125361
Right answer, n = 10, ans = 1095843
Right answer, n = 11, ans = 11954649
Right answer, n = 12, ans = 126009549
Right answer, n = 13, ans = 1628431107
Right answer, n = 14, ans = 20319845325
15 tests complited with 0 errors
** Note: $stop : ./tb/auto_testbench.v(65)
Time: 750700 ns Iteration: 1 Instance: /auto_testbench

فایل های مربوط به تمرین ۱(مسیر داده و کنترلر)

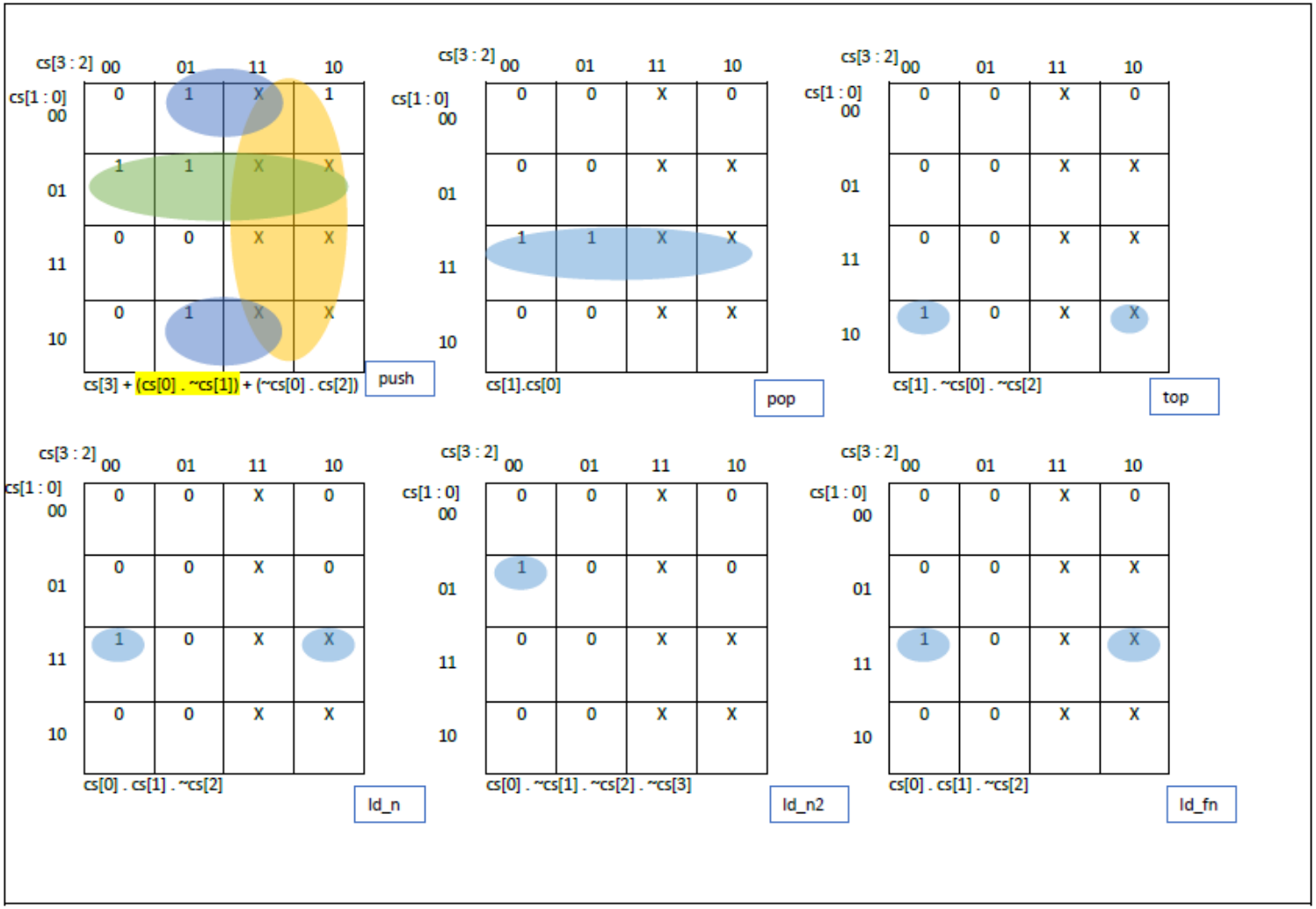
مسیر داده نهایی(بدون تغییر)



استیت دیاگرام نهایی(بدون تغییر)



کارنومپ های نهایی(با تغییر اندک)



بیضی های همرنگ در هر جدول به هم متصل می‌باشند.

cs[3 : 2] / cs[1 : 0]	00	01	11	10
00	X	1	X	0
01	0	0	X	X
11	X	X	X	X
10	X	0	X	X

~cs[0] . ~cs[1] . cs[2]

sl2

cs[3 : 2] / cs[1 : 0]	00	01	11	10
00	X	X	X	1
01	1	0	X	X
11	X	X	X	X
10	X	0	X	X

cs[3] + (~cs[3] + ~cs[2])

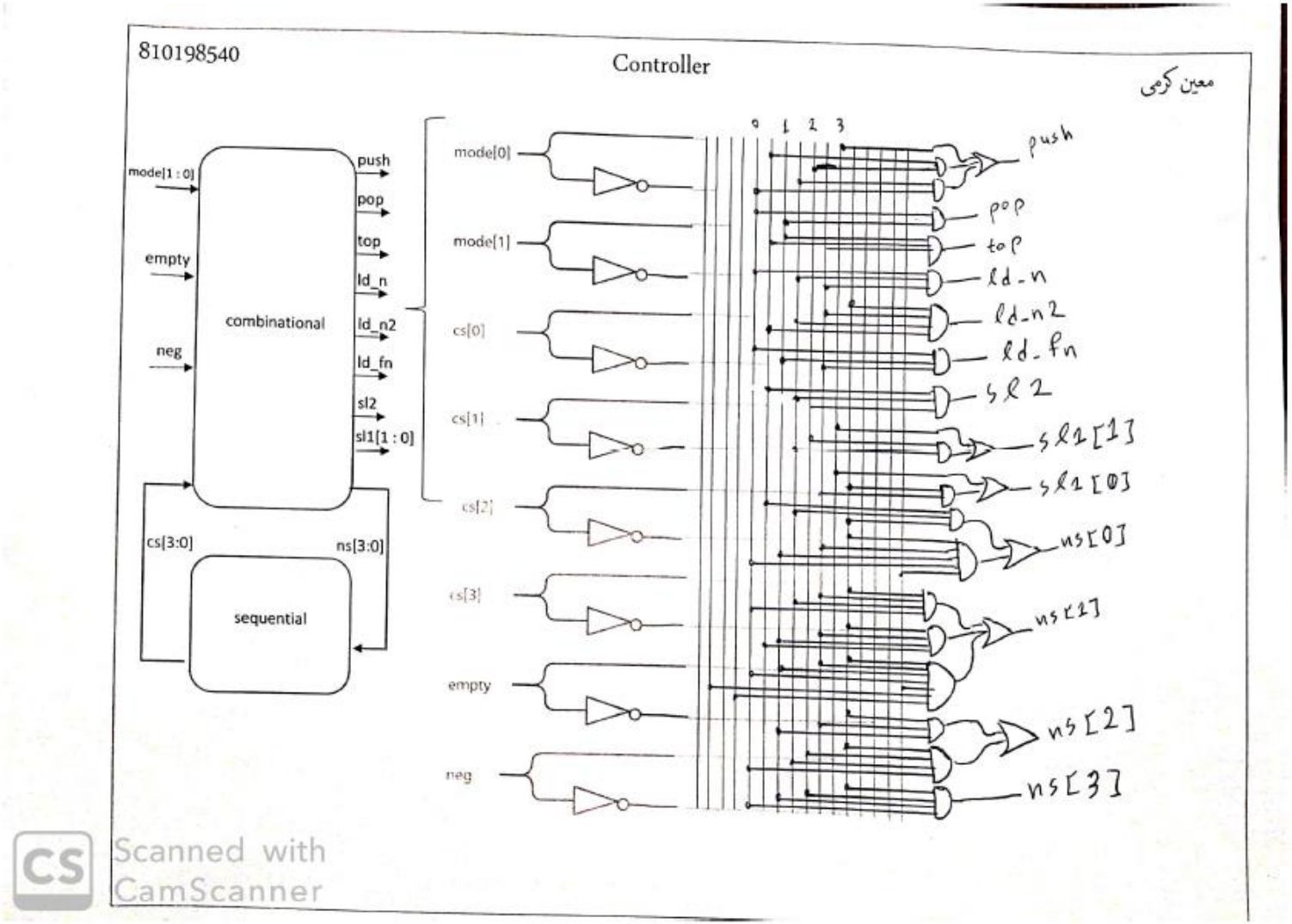
sl1[0]

cs[3 : 2] / cs[1 : 0]	00	01	11	10
00	X	X	X	1
01	0	1	X	X
11	X	X	X	X
10	X	0	X	X

cs[3] + (~cs[1] . cs[2])

sl1[1]

طراحی گیت‌لول کنترلر(بدون تغییر)



مشکلات طراحی در تمرین اول (سید محمد امین اطیابی) :

به طور کلی طراحی که برای تمرین اول انجام شد از لحاظ مسیر داده و کنترلر پیچیدگی بیسار زیادی داشت و همین امر پیاده سازی و اشکال زدایی را بسیار سخت می‌کرد . از ایرادت وارد به طراحی مسیر داده می‌توان به استفاده بدون محدودیت از منابع کرد . همچنین این مورد باعث پیچیدگی اتصالات بوده است.

بدلیل تفکری که این طراحی دنبال می‌کرد (ساخت درخت فیبوناچی به طور کامل) از نظر کنترلر دارای تعداد بساز زیادی وضعیت بوده است به طوری که برای مثال به ازای هر بار فراخوانی تابع ، هر دو زیر درخت به پشته اضافه می‌شدند و با محاسبه فراخوانی روی پشته باید یک عملیات swap انجام می‌شد تا به فراخوانی تابع زیرین دسترسی یافت و زیر درخت دیگر نیز محاسبه شود .

کنترلر این طراحی به صورت رفتاری پیاده سازی شده بود که این امر جهت پیاده سازی سخت افزاری نسبت به طراحی در سطح گیت از سرعت کمتر و منبع مورد نیاز بیشتری برخوردار بود و بهینه نبود .