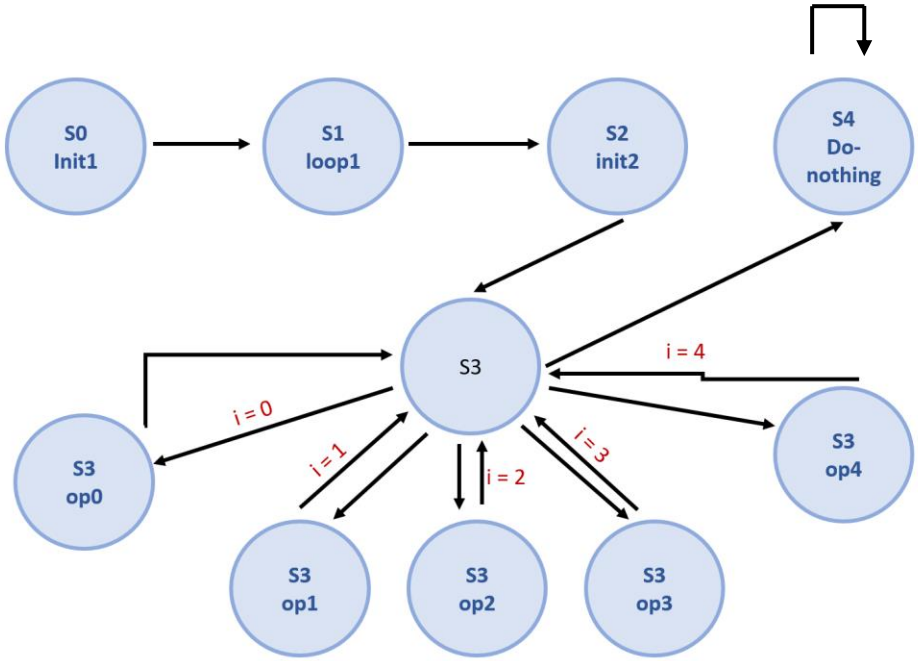
تمرین دوم هم طراحی سخت افزار – نرم افزار

نیلوفر مرادی جم 97243063

کیمیا صدیقی 97243046

سوال اول



Init1: i <- 0

loop1: taps[0 : 3] <- taps[1 : 4]

init2: taps[4] <- a, r <- 0, i <- 0

op[i]: r <- r + taps[i] * c[i], i <- i+1

سوال پنجم

گزارش کار:

اولین چالش مطرح شده برای انجام این تمرین، نصب ابزار بود. با توجه به اینکه ما اوبونتوی 64 بیتی داشتیم و gezel تنها روی اوبونتوی 32 بیتی کار میکرد، ابتدا سعی کردیم راه حلی بیابیم که نسحه 64 بیتی ما، برنامه های 32 بیتی را نیز پشتیبانی کند. با وجود اینکه ظاهرا این مسئله حل شده بود همچنان gezel نصب نمیشد.

با راهنمایی یکی از بچه ها متوجه شدیم سابقا آقای محمد صادق دهقان یک ایمیج داکر برای gezel نوشته است و به کمک داکر موفق به دریافت آن و نصب gezel شدیم.

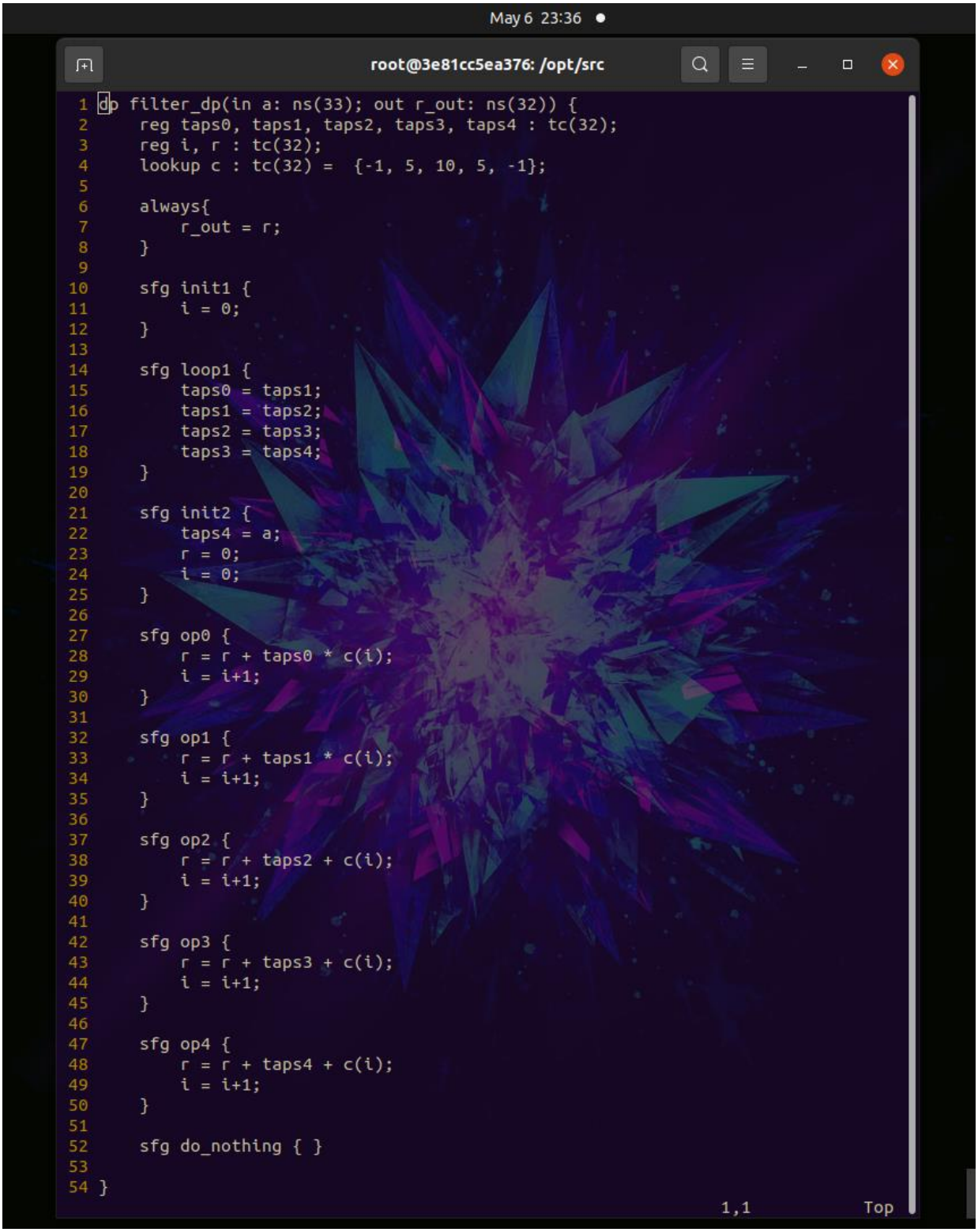
در ادامه برای نوشتن کد، فلوچارت و استیت ماشین مناسبی مطابق با صورت سوال طراحی کردیم و سعی کردیم کد آن را بنویسیم.

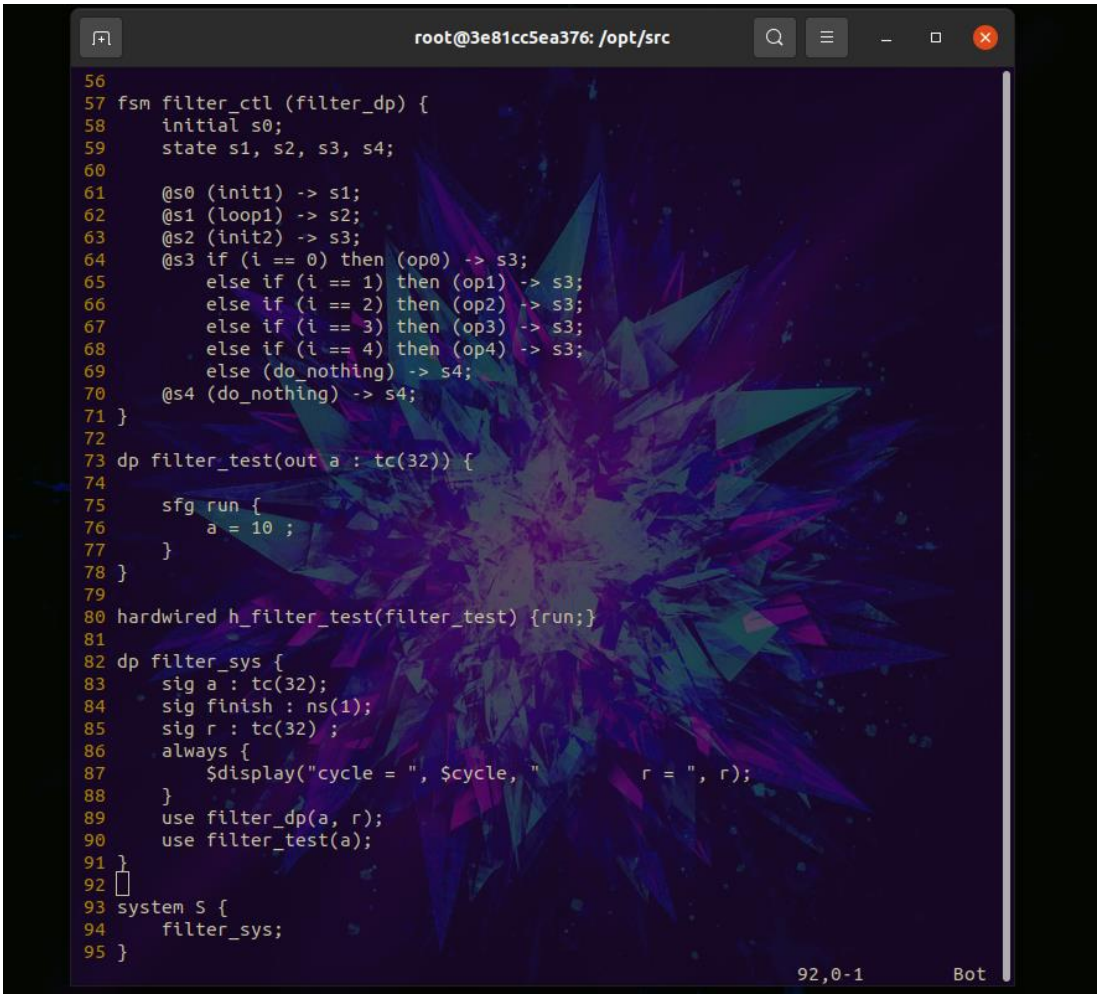
اولین چالشی که در نوشتن کد با آن روبرو شدیم، اجبار مقداردهی اولیه برای nslookup که از آن برای نوشتن آرایه taps استفاده کرده بودیم، بود.

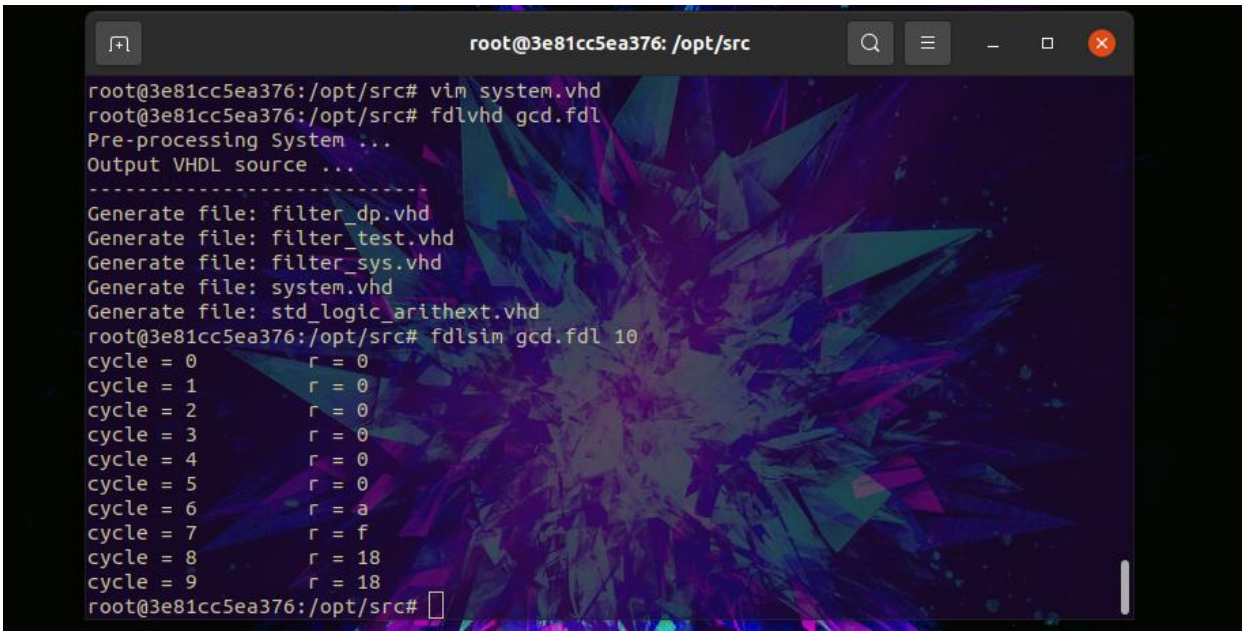
لذا برای جلوگیری از این مقداردهی اولیه، مجبور شدیم هر taps(i) را بصورت رجیستر تعریف کنیم.

چالش بعدی این بود که موفق نمیشدیم به خانه i از nslookup دسترسی پیدا کنیم. C[i] خطای سینتکسی میداد و تنها مرجع ما که کتاب و راهنمای جزل بود هم به همین روش به خانه های آرایه دسترسی داشت. با آزمون و خطا و بررسی فراوان فهمیدیم که باید از c(i) به جای c[i] استفاده کنیم.

کدها و خروجی های سوال 7:




root@3e81cc5ea376: /opt/src
56
57 fsm filter_ctl (filter_dp) {
58   initial s0;
59   state s1, s2, s3, s4;
60
61   @s0 (init1) -> s1;
62   @s1 (loop1) -> s2;
63   @s2 (init2) -> s3;
64   @s3 if (i == 0) then (op0) -> s3;
65   else if (i == 1) then (op1) -> s3;
66   else if (i == 2) then (op2) -> s3;
67   else if (i == 3) then (op3) -> s3;
68   else if (i == 4) then (op4) -> s3;
69   else (do_nothing) -> s4;
70   @s4 (do_nothing) -> s4;
71 }
72
73 dp filter_test(out a : tc(32)) {
74
75   sfg run {
76   a = 10 ;
77   }
78 }
79
80 hardwired h_filter_test(filter_test) {run;}
81
82 dp filter_sys {
83   sig a : tc(32);
84   sig finish : ns(1);
85   sig r : tc(32) ;
86   always {
87         $display("cycle = ", $cycle, "   r = ", r);
88   }
89   use filter_dp(a, r);
90   use filter_test(a);
91 }
92
93 system S {
94   filter_sys;
95 }
92,0-1 Bot


root@3e81cc5ea376: /opt/src
root@3e81cc5ea376:/opt/src# vim system.vhd
root@3e81cc5ea376:/opt/src# fdlvhd gcd.fdl
Pre-processing System ...
Output VHDL source ...

Generate file: filter_dp.vhd
Generate file: filter_test.vhd
Generate file: filter_sys.vhd
Generate file: system.vhd
Generate file: std_logic_arithext.vhd
root@3e81cc5ea376:/opt/src# fdlsim gcd.fdl 10
cycle = 0   r = 0
cycle = 1   r = 0
cycle = 2   r = 0
cycle = 3   r = 0
cycle = 4   r = 0
cycle = 5   r = 0
cycle = 6   r = a
cycle = 7   r = f
cycle = 8   r = 18
cycle = 9   r = 18
root@3e81cc5ea376:/opt/src#

کدها و خروجی های سوال 8: