محمد حسین جوادی ۸۱۰۱۰۳۵۶۷

ورودی ها بصورت S_0,S_1,S_2,S_3

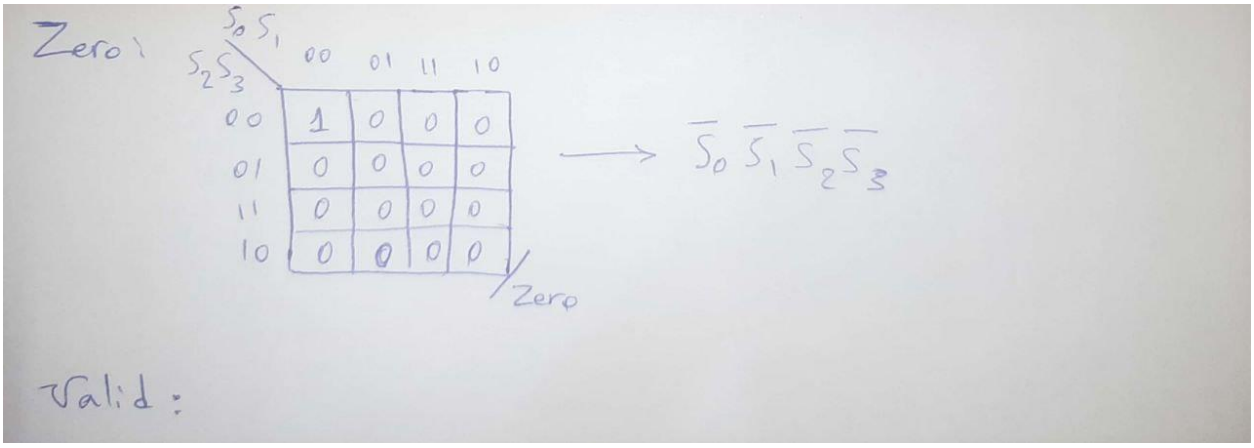
S_0:MSB

S_3:LSB

بخش ۱)

الف)

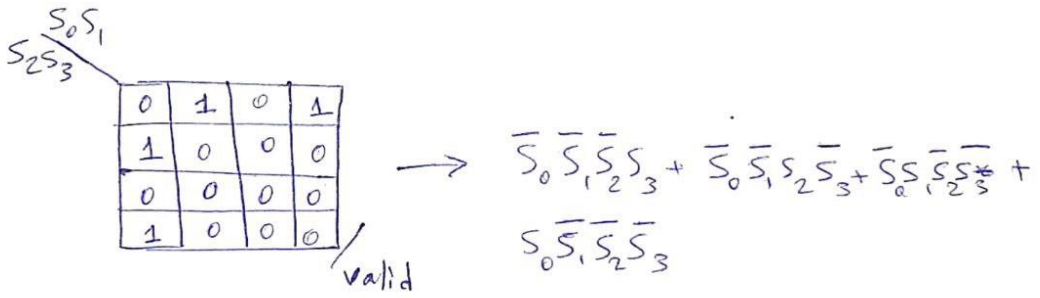
Zero:



SOP: $\sim S_0.\sim S_1.\sim S_2.\sim S_3$

Valid:

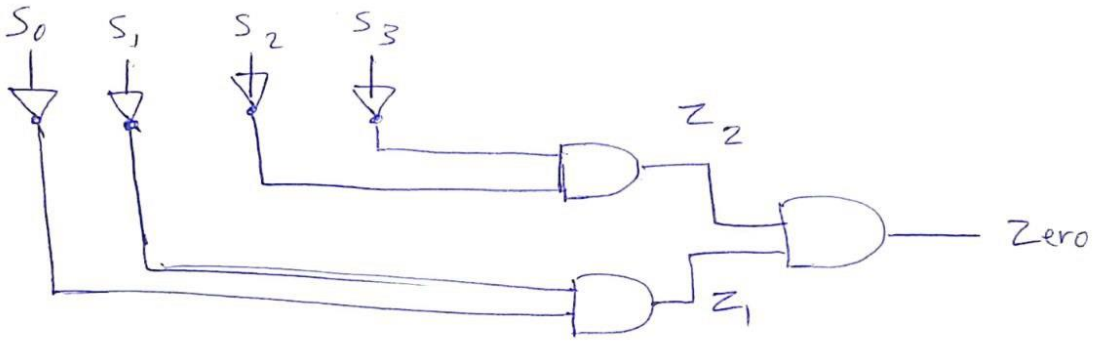
Valid :



SOP: $\sim S_0.\sim S_1.\sim S_2.S_3 + \sim S_0.\sim S_1.S_2.\sim S_3 + \sim S_0.S_1.\sim S_2.\sim S_3 + S_0.\sim S_1.\sim S_2.\sim S_3$

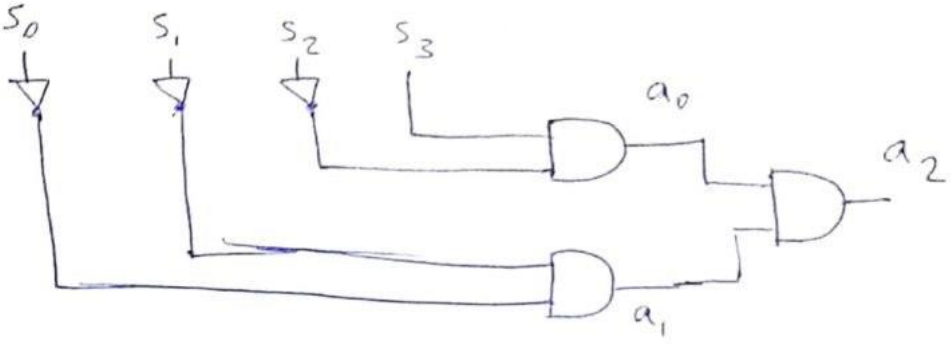
ب)

Zero: 0000

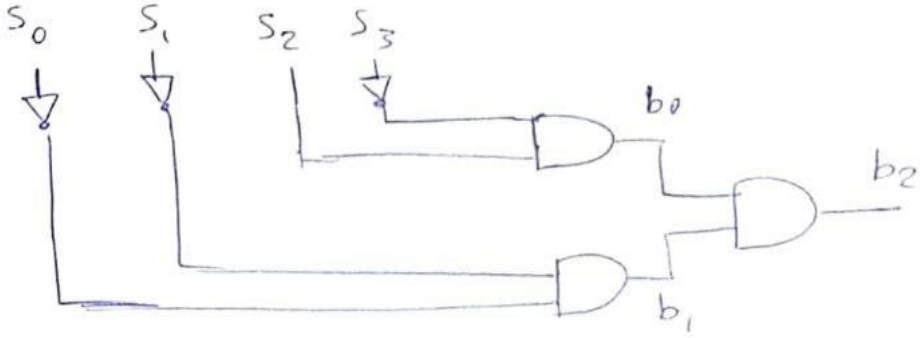


Valid:

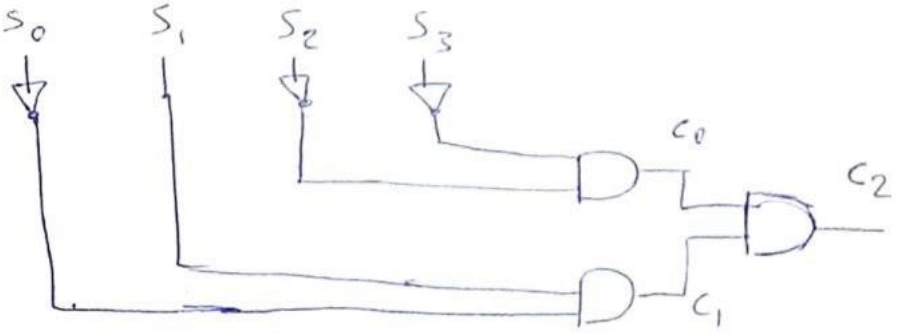
0001



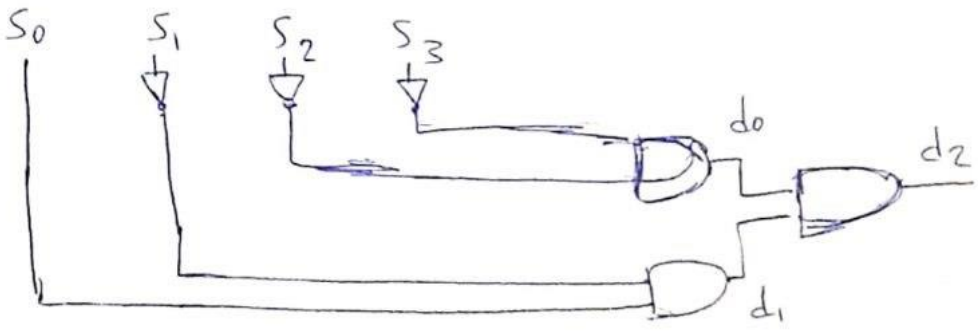
0010

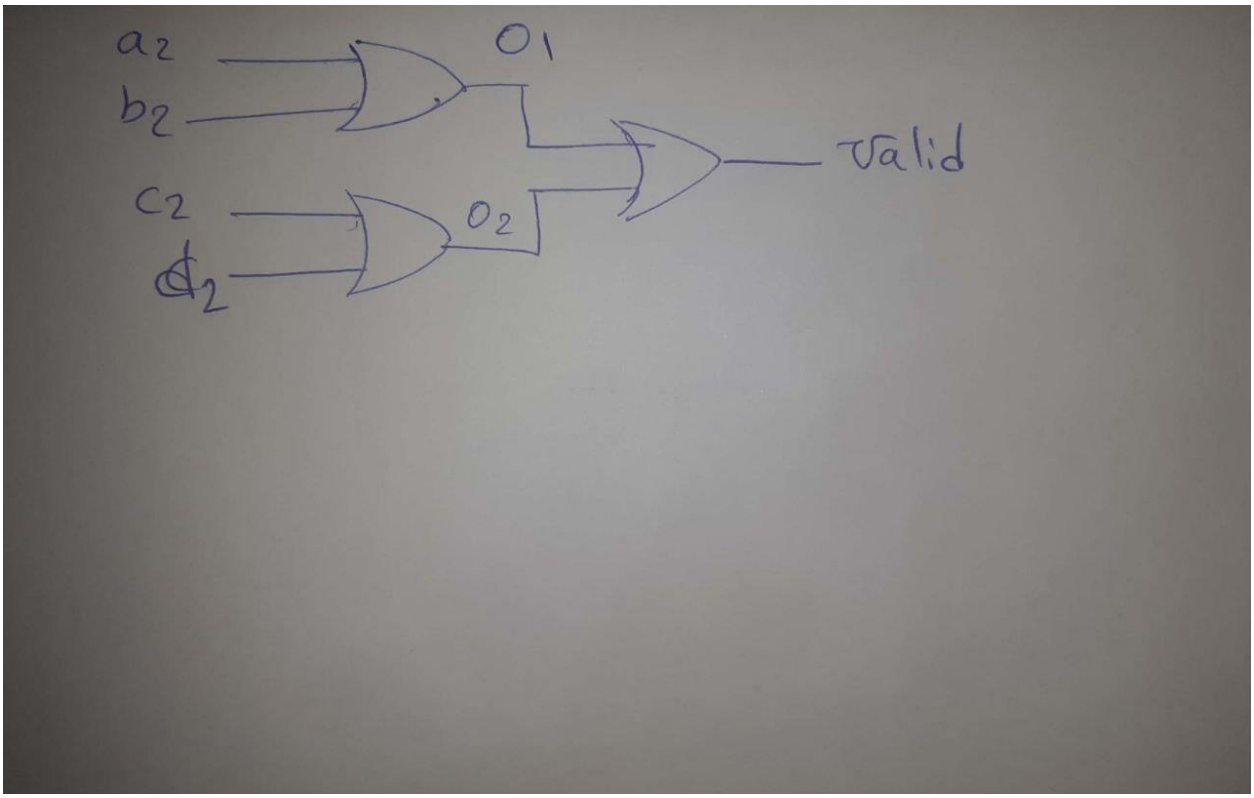


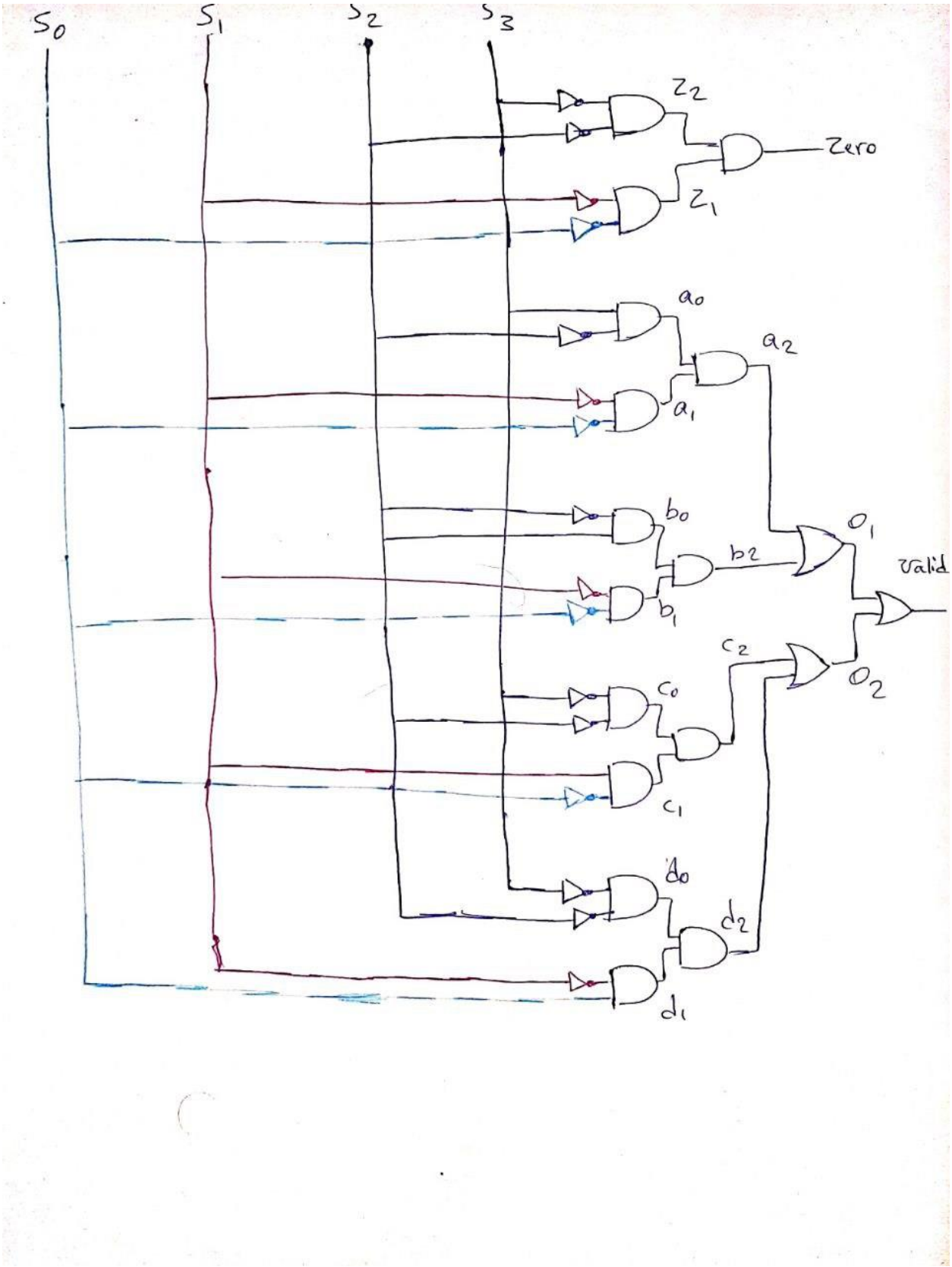
0100:



1000




a2
b2
c2
d2
O1
O2
valid


S0
S1
S2
S3
Z2
Zero
Z1
a0
a2
a1
b0
O1
b2
valid
b1
c2
O2
c0
c1
d0
d2
d1

ج)

خروجی Zero:

برابر است با ۱+۳+۳ که برابر است با ۷ نانوثانیه به این دلیل که دو گیت and و یک گیت not داریم.

خروجی Valid :

اگر از یک اینورتر، دو اند و دو اور عبور کنیم حداکثر خروجی برابر است با ۱+۳+۳+۲+۲ = ۱۱. پس حداکثر تاخیر برابر است با ۱۱ نانوثانیه.

بخش ۲)

Module Code:

```
1. `timescale 1ns/1ns
2.
3. module OnehotDetector (input [3:0]S,output Zero,output Valid);
4.
5. wire n0,n1,n2,n3;
6.
7. not #(1) s0_not(n0,S[0]);
8. not #(1) s1_not(n1,S[1]);
9. not #(1) s2_not(n2,S[2]);
10. not #(1) s3_not(n3,S[3]);
11.
12. wire z1,z2;
13.
14. and #(3) zero_first(z1,n0,n1);
15. and #(3) zero_second(z2,n2,n3);
16. and #(3) zero_third(Zero,z1,z2);
17.
18. wire a0,a1,a2;
19.
20. and #(3) first_1_and(a0,S[3],n2);
21. and #(3) second_1_and(a1,n1,n0);
22. and #(3) third_1_and(a2,a1,a0);
23.
24. wire b0,b1,b2;
25.
26. and #(3) first_2_and(b0,n3,S[2]);
27. and #(3) second_2_and(b1,n0,n1);
28. and #(3) third_2_and(b2,b1,b0);
29.
30. wire c0,c1,c2;
31.
32. and #(3) first_4_and(c0,n3,n2);
33. and #(3) second_4_and(c1,n0,S[1]);
34. and #(3) third_4_and(c2,c1,c0);
35.
36. wire d0,d1,d2;
37.
38. and #(3) first_8_and(d0,n3,n2);
```

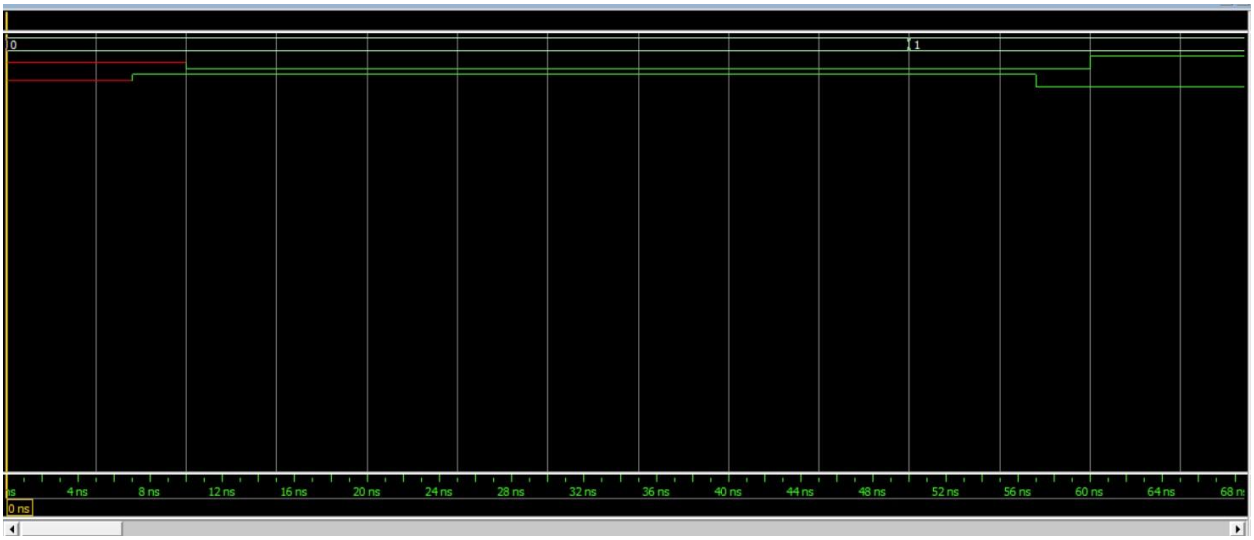
```
39. and #(3) second_8_and(d1,S[0],n1);
40. and #(3) third_8_and(d2,d1,d0);
41.
42. wire o1,o2;
43.
44. or #(2) first_or(o1,a2,b2);
45. or #(2) second_or(o2,c2,d2);
46. or #(2) last_or(Valid,o1,o2);
47.
48. endmodule
49.
```

بخش ۳)

Testbench Code:

```
1. `timescale 1ns/1ns
 2.
 3. module test_bench();
 4.
 5. logic [3:0]S;
 6. wire Zero;
 7. wire Valid;
 8.
 9. OnehotDetector inst(.S(S),.Zero(Zero),.Valid(Valid));
10.
11. initial begin
12.     S = 4'd0;
13.
14.     #50 S = 4'd1;
15.     #50 S = 4'd2;
16.     #50 S = 4'd3;
17.     #50 S = 4'd4;
18.     #50 S = 4'd5;
19.     #50 S = 4'd6;
20.     #50 S = 4'd7;
21.     #50 S = 4'd8;
22.     #50 S = 4'd9;
23.     #50 S = 4'd10;
24.     #50 S = 4'd11;
25.     #50 S = 4'd12;
26.     #50 S = 4'd13;
27.     #50 S = 4'd14;
28.     #50 S = 4'd15;
29.     #50 $stop;
30.
31. end
32.
33. endmodule;
34.
35.
```

ورودی ۰ و ۱:



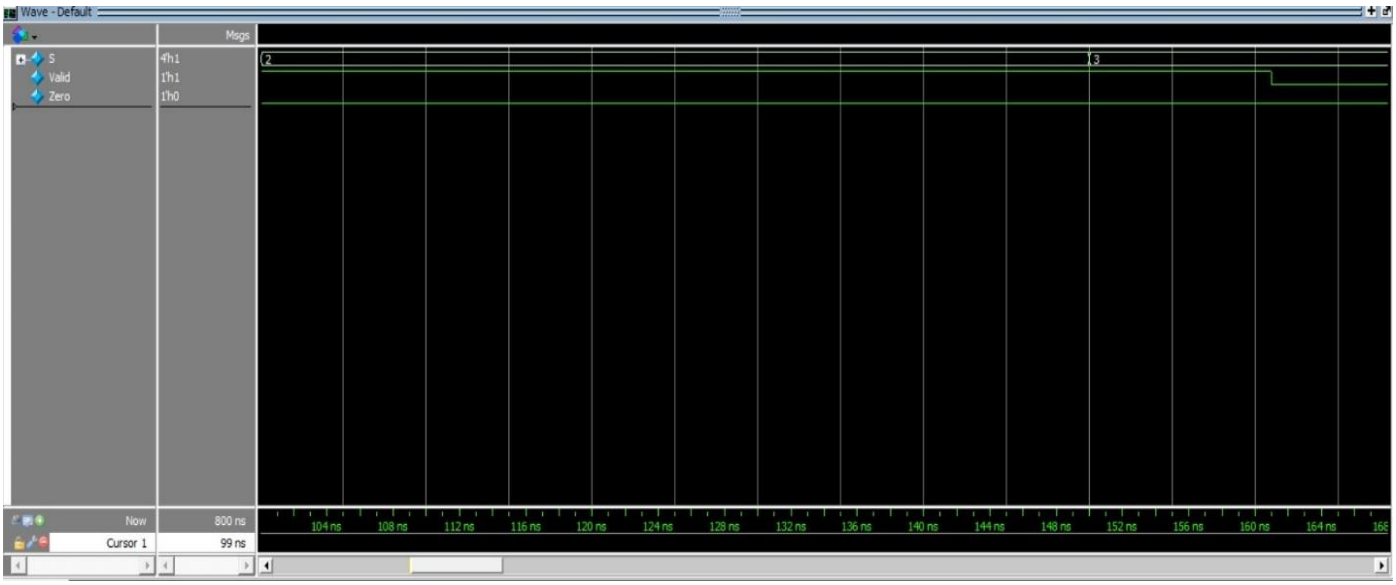
ورودی ۰ :

خروجی Zero برابر 1 و تاخیر ان 7ns(همواره به این دلیل که همه ورودی ها اینورتر دارند و از دو اند میگذرند) و برای Valid برابر 0 و تاخیر ان 10ns است چرا که برای Zero هر ورودی از یک اینورتر و دو اند میگذرند و برای valid در هر and یک سیم بدون اینورتر داریم که باعث میشود

در 10ns خروجی بدهد. همواره پس از 11ns خروجی براساس تمام ورودی ها خواهد بود.

ورودی ۱:

خروجی Zero برابر 0 و تاخیر ان 7ns (همواره به این دلیل که همه ورودی ها اینورتر دارند و از دو اند میگذرند) و برای valid برابر 1 و تاخیر ان برابر 10ns چرا که S3 بدون اینورتر وارد گیت اند میشود و باعث میشود 1ns کم تر طول بکشد برای خروجی. همواره پس از 11ns خروجی براساس تمام ورودی ها خواهد بود.

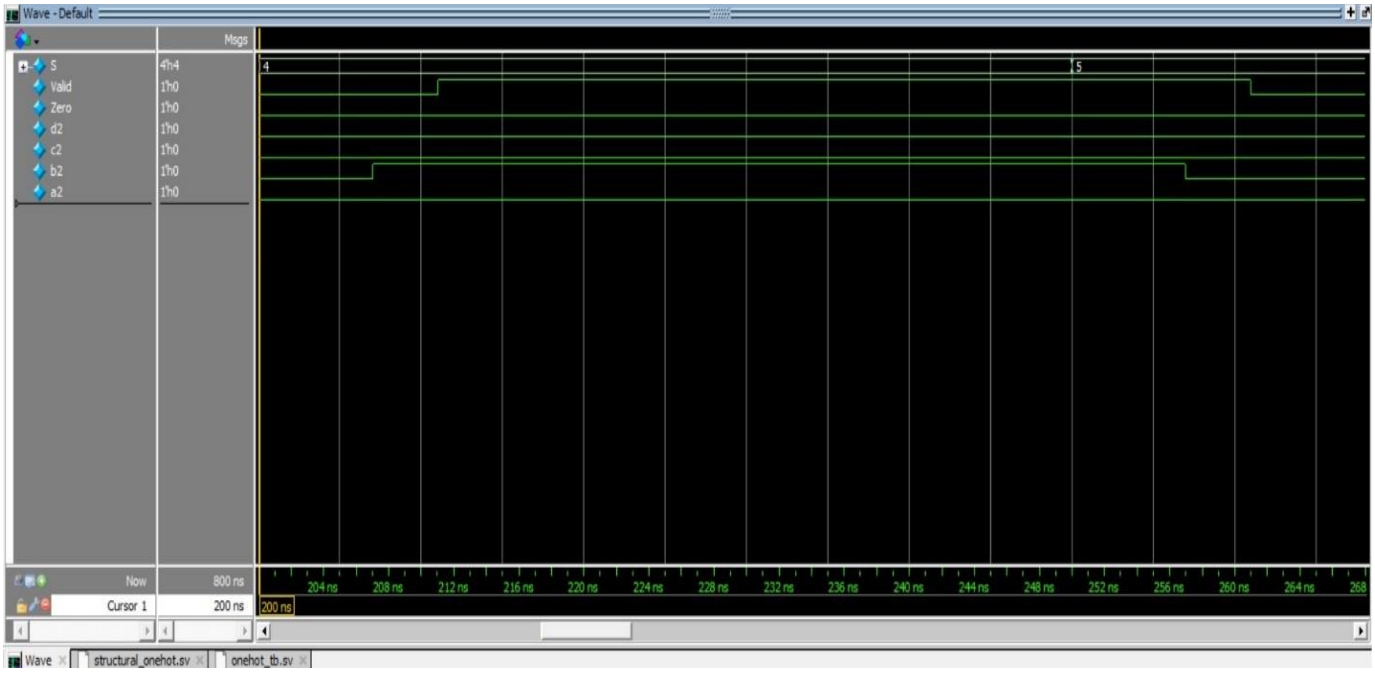


ورودی ۲:

خروجی Zero برابر 0 و تاخیر ان 7ns(همواره به این دلیل که همه ورودی ها اینورتر دارند و از دو اند میگذرند) و برای Valid برابر 1 است و تاخیر آن 11ns است چرا که b2 پس از 7ns مقدارش ۱شده پس با دو اور برابر 11ns میشود.

ورودی ۳:

خروجی Zero برابر 0 و تاخیر ان 7ns (همواره به این دلیل که همه ورودی ها اینورتر دارند و از دو اند میگذرند) و برای Valid برابر 0 و تاخیر ان 11ns است. چرا که c2پس از 7ns برابر ۰ میشود و پس از گذر از دو اور برابر 11ns میشود.

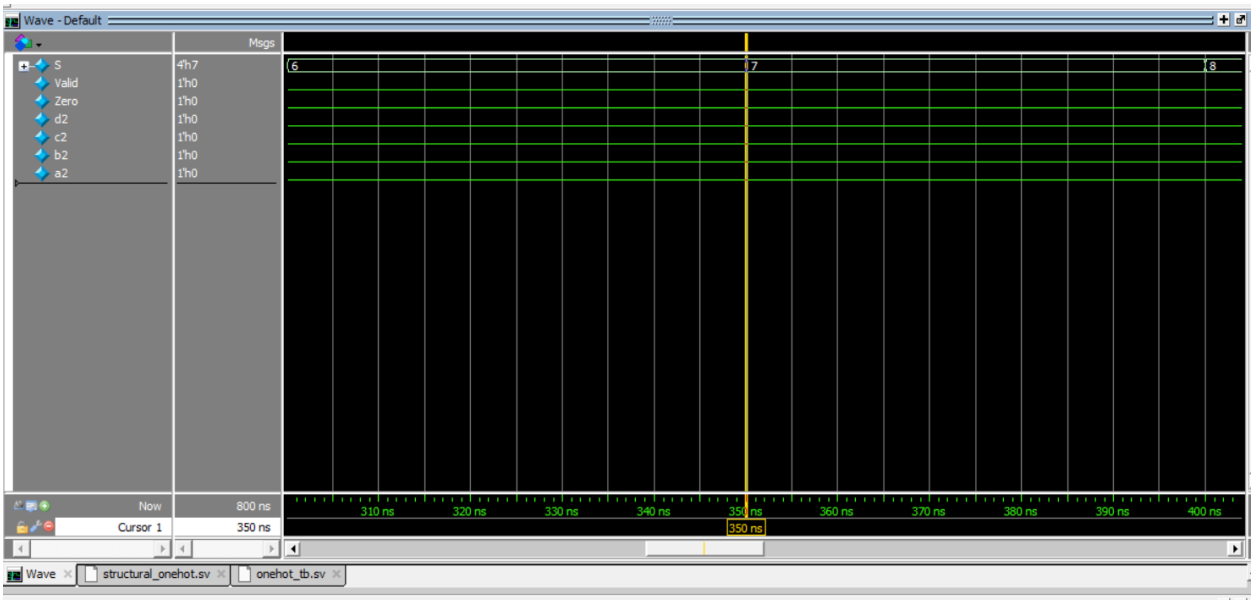


ورودی ۴:

خروجی Zero برابر 0 و تاخیر ان 7ns (همواره به این دلیل که همه ورودی ها اینورتر دارند و از دو اند میگذرند) و برای Valid برابر 1 است و تاخیر آن 11ns است. چرا که بعد از 7ns اند b2 برابر 1 میشود و پس از دو اور 4ns ایی میشود 11ns.

ورودی ۵:

خروجی Zero برابر 0 و تاخیر ان 7ns (همواره به این دلیل که همه ورودی ها اینورتر دارند و از دو اند میگذرند) و Valid برابر 0 است و تاخیر آن برابر 11ns است چرا که b2 پس از 7ns برابر 0 میشود و بقیه اند ها هم صفر هستند و پس از دو اور میشود 11ns.

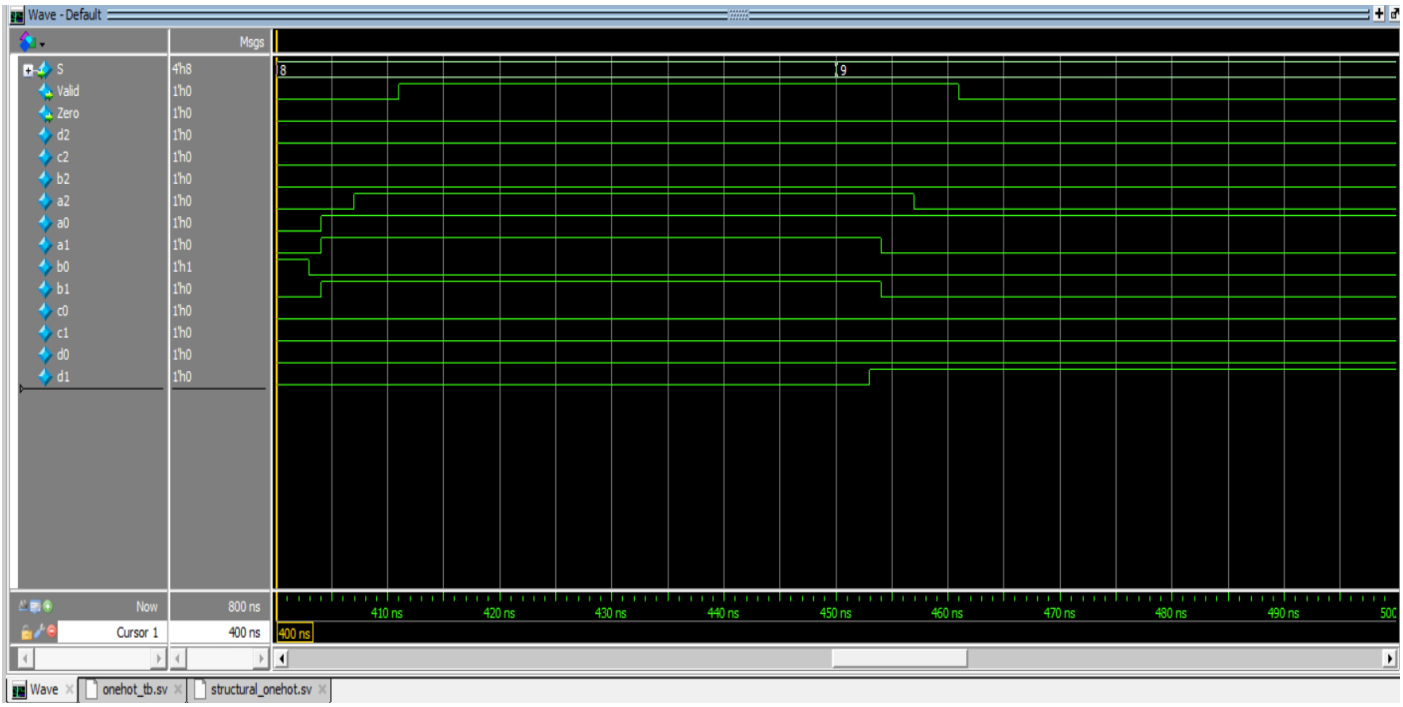


ورودی ۶:

خروجی Zero برابر 0 و تاخیر ان 7ns و خروجی valid برابر است با 0 و تاخیر آن 10ns چرا که تمامی اند ها همچنان صفر میمانند پس از d1 3ns برابر 1 و پس از دو اور و یک اند میشود 10ns .

ورودی ۷:

خروجی Zero برابر 0 و تاخیر ان 7ns و خروجی Valid برابر 0 و تاخیر ان 11ns چرا که c1 پس از 4ns صفر میشود و پس از از دو اور و یک اند میشود 11ns .(بقیه اند های لول دو صفرند)

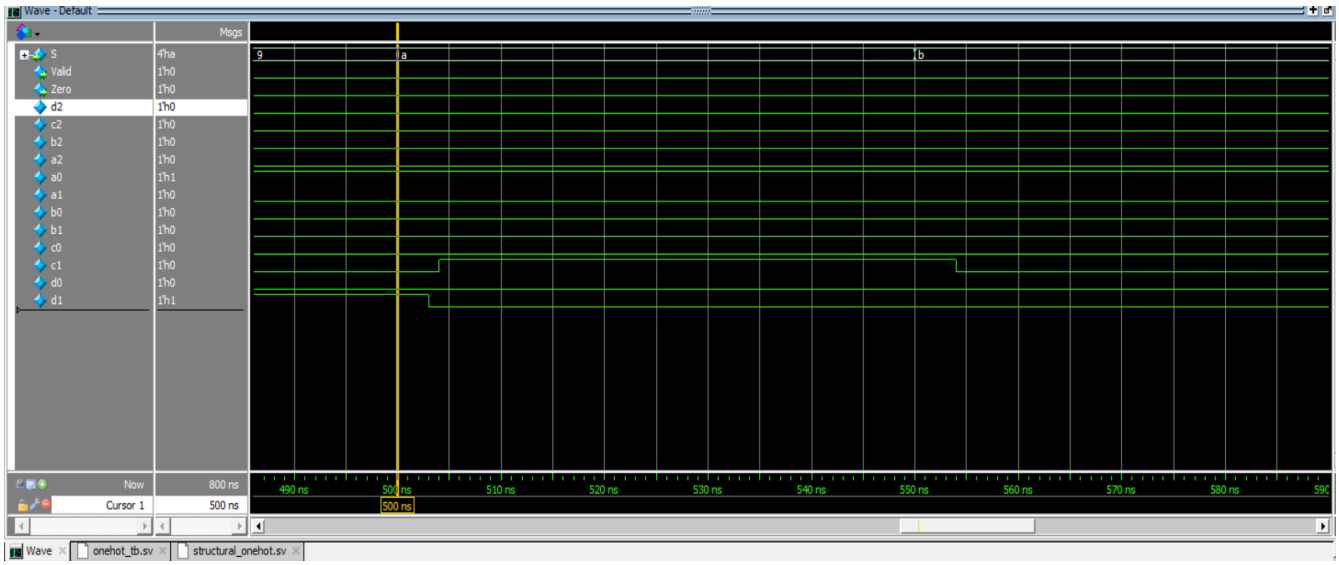


ورودی ۸:

خروجی Zero برابر 0 و تاخیر ان 7ns و برای Valid برابر 1 است و تاخیر آن 11ns است. چرا که a2 پس از 7ns یک میشود و از انجایی که اور میشود خروجی برابر 1 میشود و به دلیل گذر از دو اور تاخیر برابر 11ns میشود.

ورودی ۹:

خروجی Zero برابر 0 و تاخیر ان 7ns و برای Valid برابر 0 است و تاخیر آن 11ns است چرا که a2 پس از 7ns برابر 0 میشود و باقی اند ها هم 0 بودند که با گذر از دو اور خروجی برابر 0 میشود و در نهایت برابر 11ns میشود.

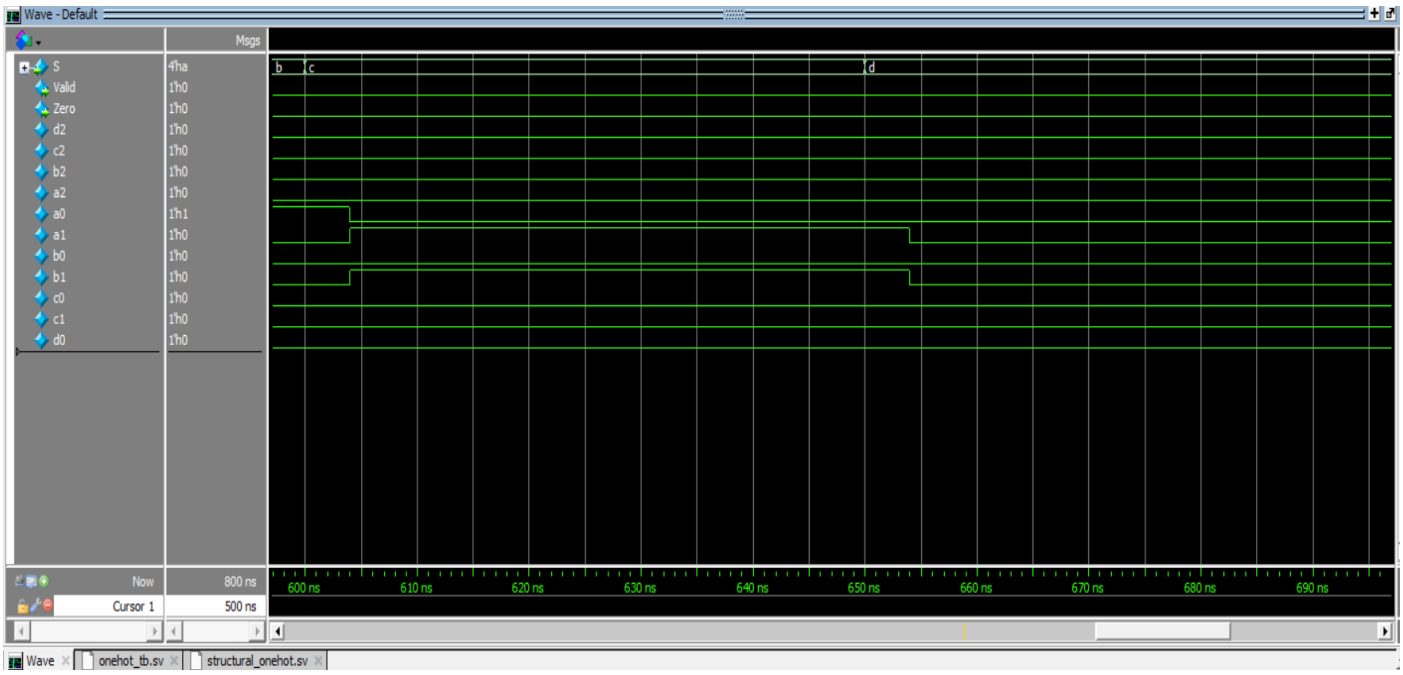


ورودی ۱۰:

خروجی Zero برابر 0 و تاخیر ان 7ns و خروجی valid برابر 0 میشود و تاخیر برابر 10ns میشود چرا که d1 پس از 3ns برابر ۱ میشود و c1 پس از 4ns برابر 1 میشود ولی تمامی اند های لول 2 همگی صفر میمانند و پس از از گذر از یک اند و دو اور همچنان 0 میماند و تاخیر نیز برابر 10ns میشود

ورودی ۱۱:

خروجی Zero برابر 0 و تاخیر ان 7ns و خروجی valid برابر 0 میشود و تاخیر ان برابر 11ns میشود چرا که c1 پس از 4ns برابر صفر میشود ولی اند های لول دو صفر میمانند در نتیجه پس از گذر از یک اند و دو اور همچنان 0 میماند و تاخیر در نهایت برابر 11nsمیشود.

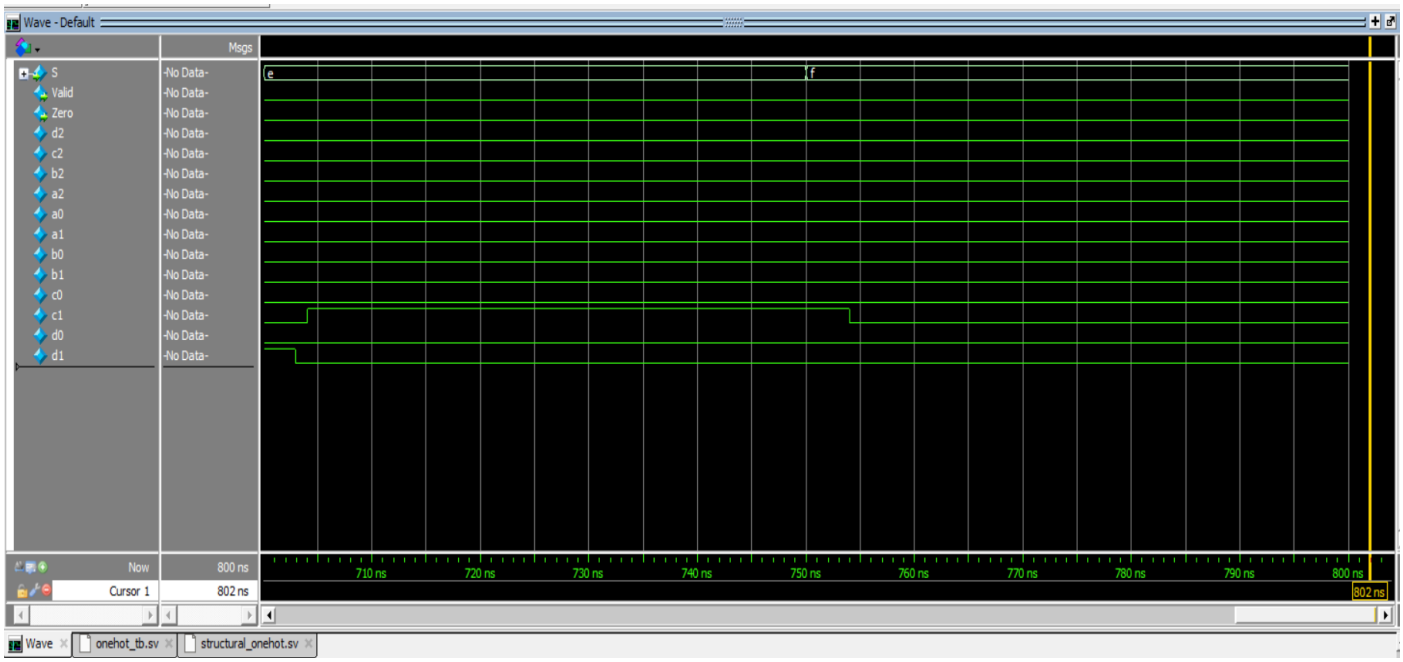


ورودی ۱۲:

خروجی Zero برابر 0 و تاخیر ان 7ns و Valid برابر 0 میشود و تاخیر ان 11ns میشود چرا که پس از 4ns a1,b1 برابر 1 و a0 برابر 0 میشود ولی اند های لول 2 تغییر نمیکنند در نتیجه پس از گذر از یک اند و دو اور در نهایت 0 میشود و تاخیر برابر 11ns میشود

ورودی ۱۳:

خروجی Zero برابر 0 و تاخیر ان 7ns و valid برابر 0 میشود و تاخیر آن 10ns میشود چرا که پس از 3ns d1 برابر 1 و پس از 4ns b1,a1 برابر 0 میشوند ولی اند های لول 2 همچنان 0 هستند پس خروجی باز 0 میشود و 3+3+2+2 =10ns



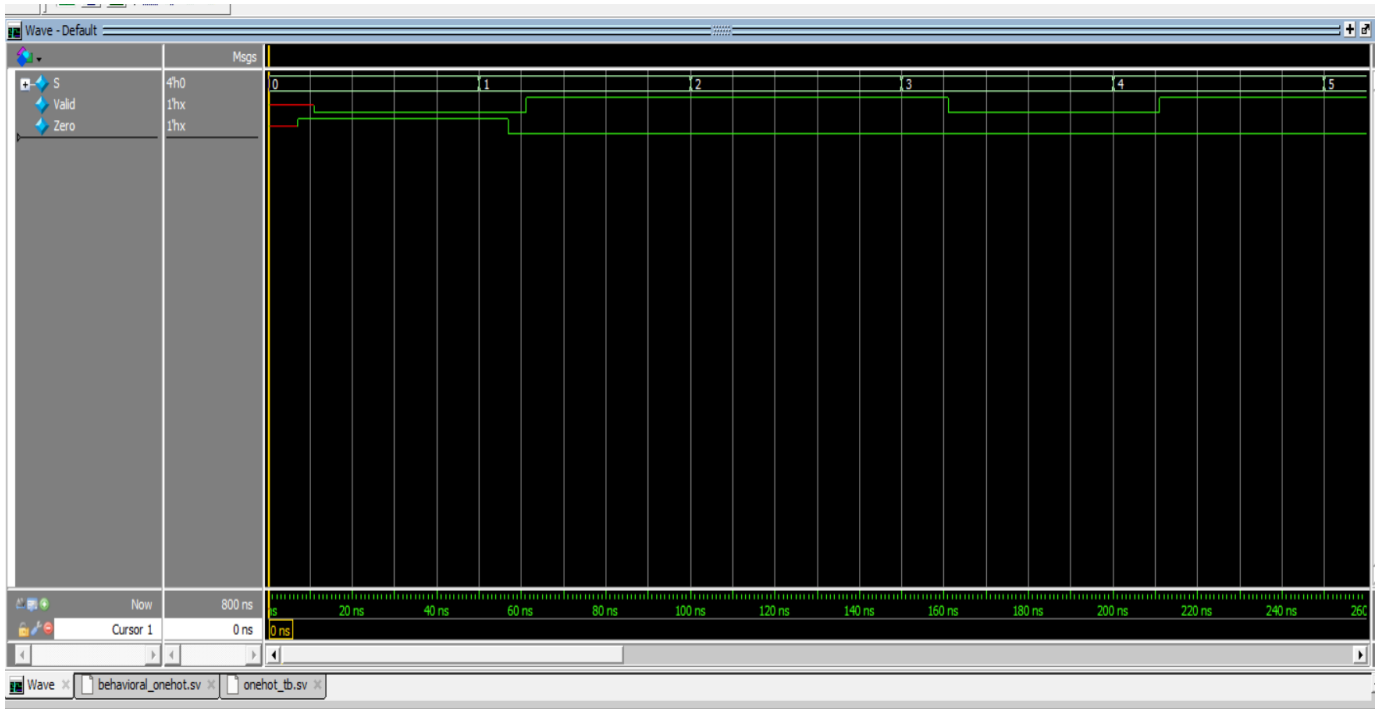
ورودی ۱۴: خروجی Zero برابر 0 و تاخیر ان 7ns و خروجی valid برابر 0 و تاخیر ان برابر 10ns چرا که پس از d1 3ns برابر 0 و c1 4ns برابر 1 میشود ولی اند های لول 2 برابر 0 هستند و در نهایت خروجی 0 میشود و 3+3+2+2 = 10ns

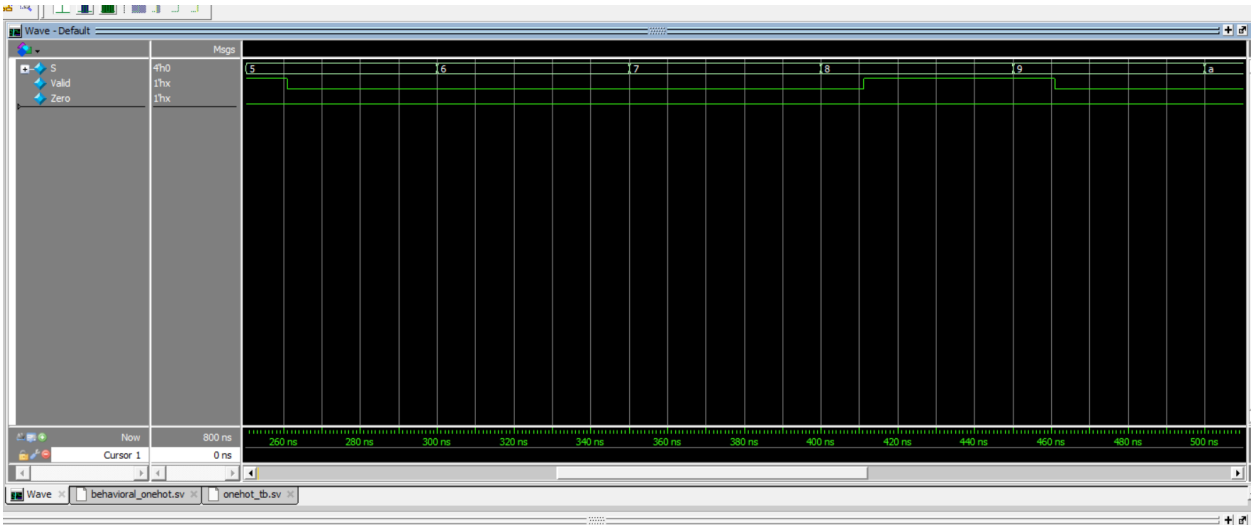
ورودی ۱۵:

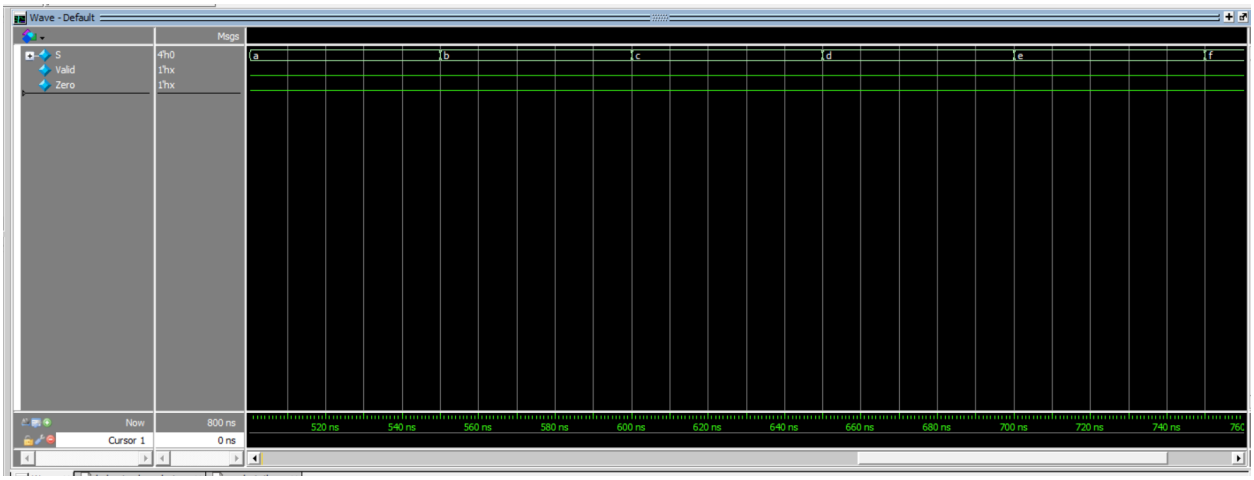
خروجی Zero برابر 0 و تاخیر ان 7ns و خروجی validبرابر 0 و تاخیر ان 11ns است چرا که پس از 4ns c1 برابر 0 میشود ولی همچنان اند های لول 2 برابر 0 هستند و خروجی در نهایت برابر 0 میشود و خروجی برابر 0 میشود و 4+3+2+2 = 11ns

بخش۴)

```
1. 1. `timescale 1ns/1ns
2. 2.
3. 3. module OnehotDetector (input [3:0]S,output Zero,output Valid);
4. 4.     assign #(7) Zero = ~S[0] & ~S[1] & ~S[2] & ~S[3];
5. 5.     assign #(11) Valid = (~S[0] & ~S[1] & ~S[2] & S[3]) | (~S[0] & ~S[1] & S[2] & ~S[3]) |
(~S[0] & S[1] & ~S[2] & ~S[3]) | (S[0] & ~S[1] & ~S[2] & ~S[3]) ;
6. 6. endmodule
7. 7.
8.
```


Wave - Default
Msgs
S
Valid
Zero
4'h0
1'hx
1'hx
0
1
2
3
4
5
Now
800 ns
Cursor 1
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
180 ns
200 ns
220 ns
240 ns
0 ns
Wave
behavioral_onehot.sv
onehot_tb.sv


Wave - Default
Msgs
S
Valid
Zero
4'h0
1'hx
1'hx
5
6
7
8
9
a
Now
800 ns
Cursor 1
0 ns
260 ns
280 ns
300 ns
320 ns
340 ns
360 ns
380 ns
400 ns
420 ns
440 ns
460 ns
480 ns
500 ns
Wave
behavioral_onehot.sv
onehot_tb.sv



در این جا باتوجه به نمودار ها میبینیم که تاخیر همه ی ورودی ها یکسان است.

در این طراحی دقت تاخیر ها کم تر میشود اما نوشتن کد راحت تر و براساس SOP است. ولی بدلیل صرف نظر از تاخیر از همه‌ی تاخیر ها دقت behavioral کم تر است. ما در structural از گیت ها استفاده میکنیم و تمام ورودی خروجی های مدار های بخش ۱ را شبیه سازی میکنیم که از دقت کافی برخوردار است ولی ممکن است طراحی ان سخت تر باشد.