

به نام خدا

تمرین دوم درس سیستم‌های نهفته بی‌درنگ



نیمسال دوم ۱۴۰۱-۱۴۰۲

سوال ۲)

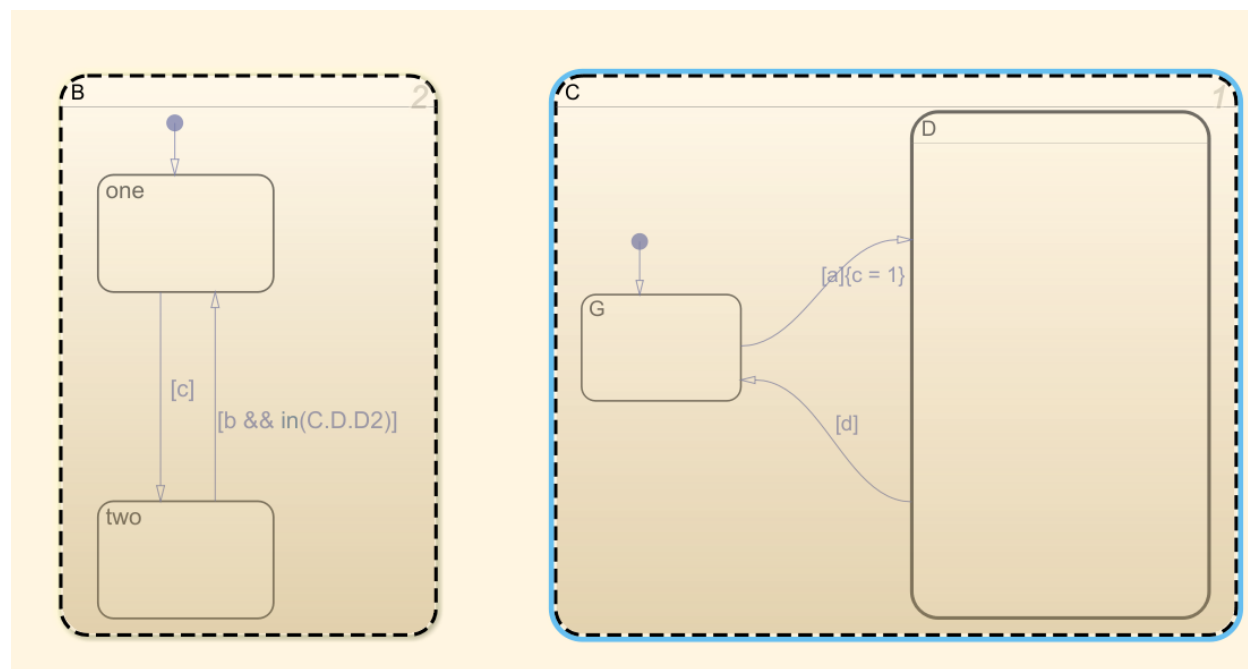
(الف)

خط چین بین B,C نشان دهنده AND است و یعنی هر اتفاقی که در سمت B بیفتد در سمت C قابل مشاهده است. حالت های اولیه برای شروع در داخل B,C حالت های 1 و G هستند.

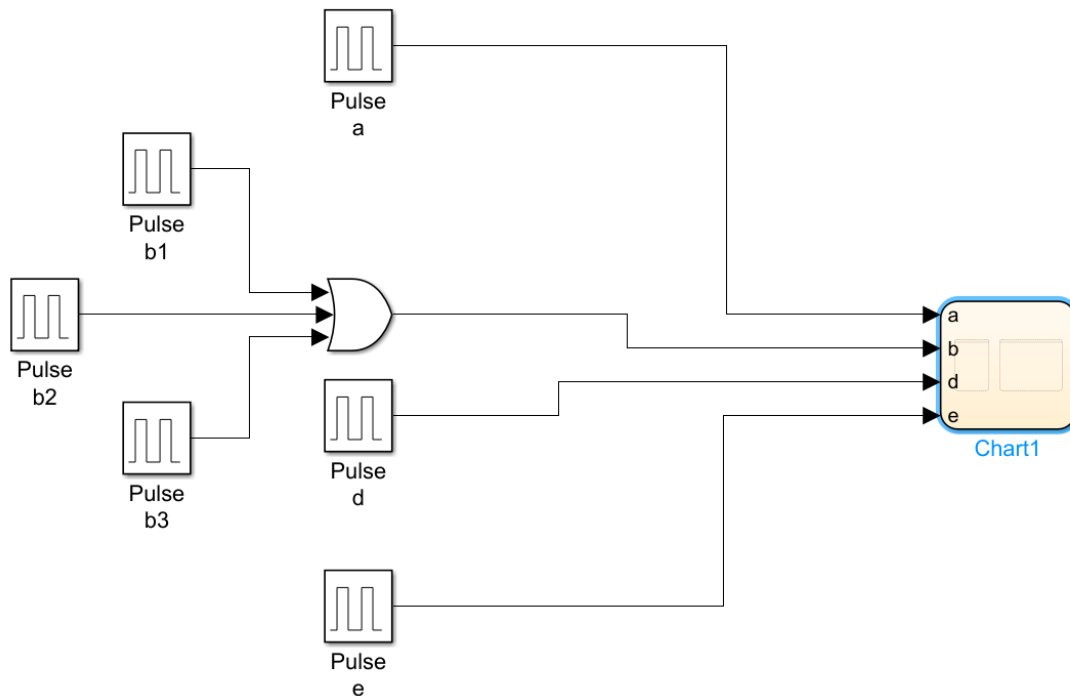
	B	C	1	2	G	D	D1	D2
Init	x	x	x		x			
a	x	x		x		x	x	
b	x	x		x		x	x	
e	x	x		x		x		x
b	x	x	x			x		x
d	x	x	x		x			
b	x	x	x		x			

(ب)

ابتدا استیت چارت را در سیمولینک رسم میکنیم:



با استفاده از تابع pulse generator ورودی های تابع را به state chart مان میدهیم.



۵ سیگنال اشاره شده در صورت سوال را روی حالت input قرار میدهیم زیرا با present شدن آن ها استیت های ما تغییر میکنند پس اینکه چه زمانی هایی حاضر باشند و چه زمان هایی غایب رفتار را تعیین میکند.

زمانبندی ورودی ها:

۱- ابتدا ورودی a با تاخیر زمانی ۲ ثانیه.

۲- ورودی b با تاخیر زمانی ۴ ثانیه

۳- ورودی e با تاخیر زمانی ۶ ثانیه.

۴- ورودی b با تاخیر زمانی ۸ ثانیه

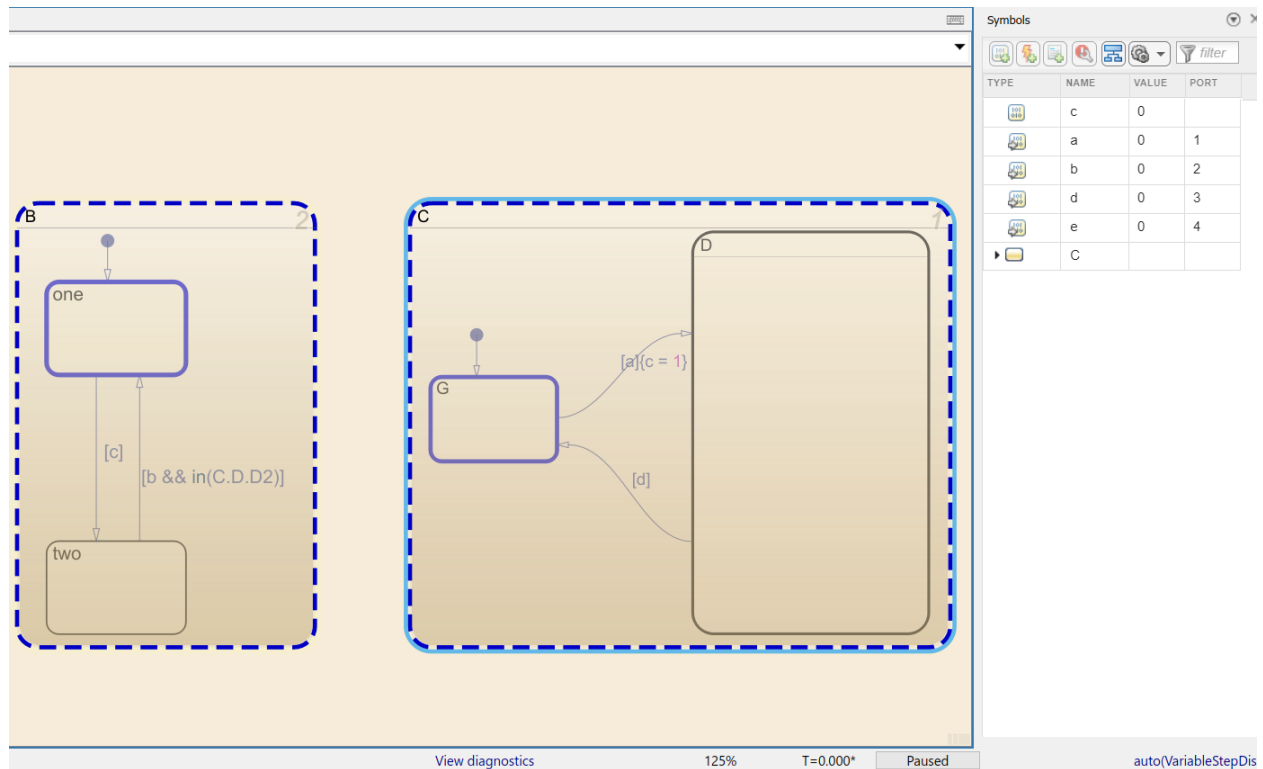
۵- ورودی d با تاخیر زمانی ۱۰ ثانیه

۶- ورودی b با تاخیر زمانی ۱۲ ثانیه.

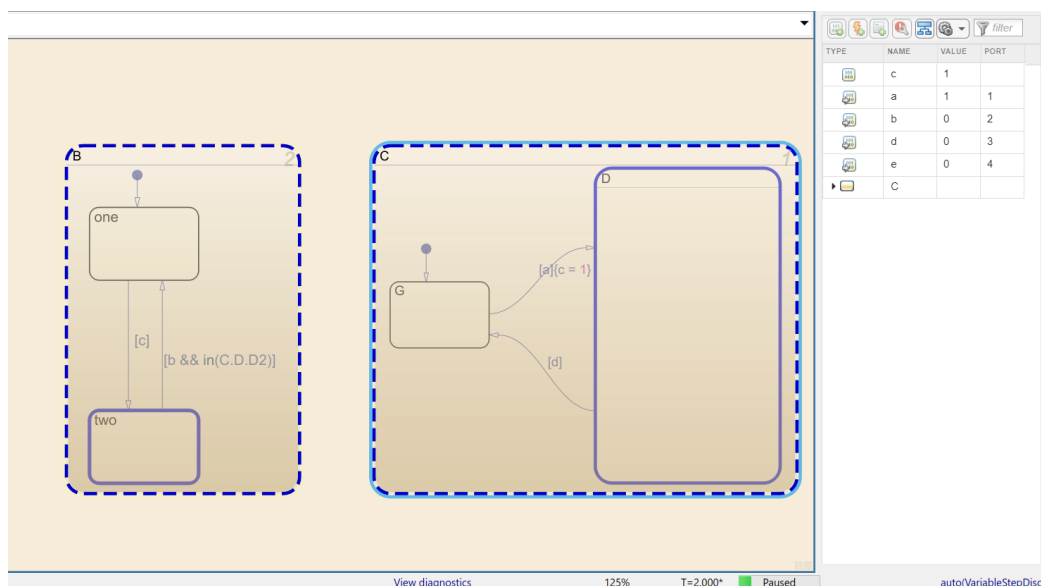
سپس در symbol view را انتخاب کرده و مقدار دهی می کنیم.

از step forward debugging بران نشان دادن درستی رفتار سیستم استفاده میکنیم:

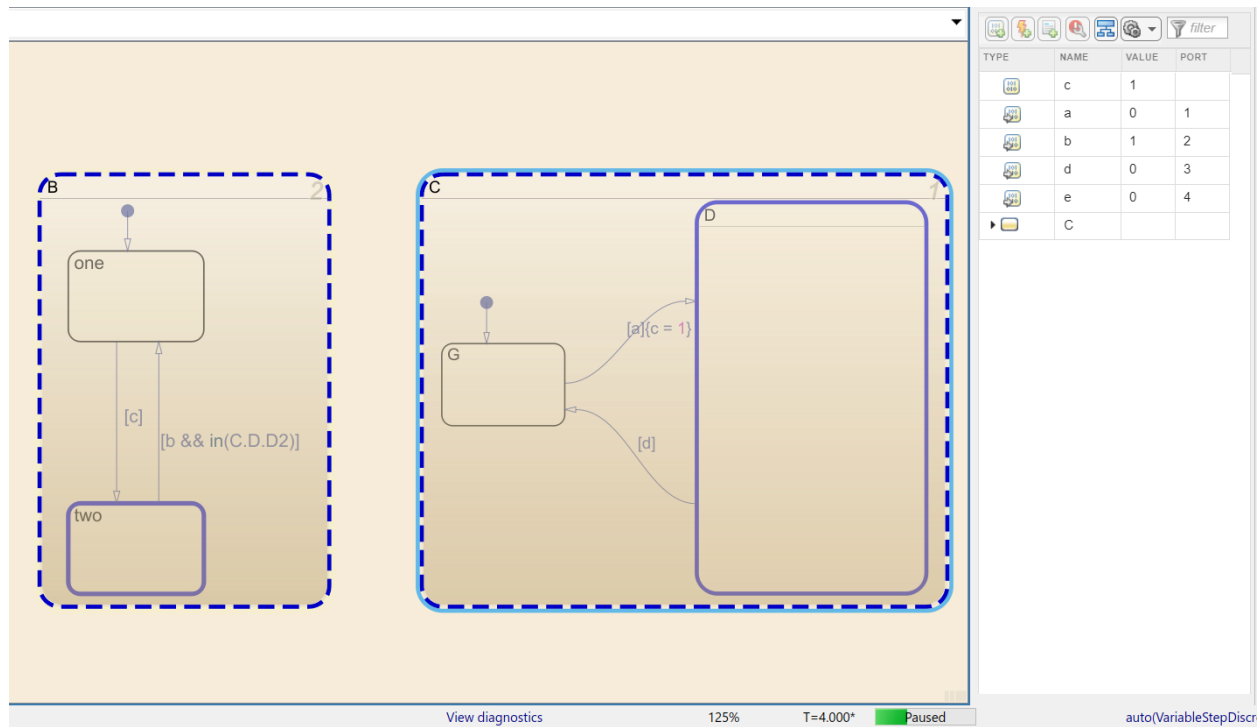
ابتدا در زمان 0 هستیم:



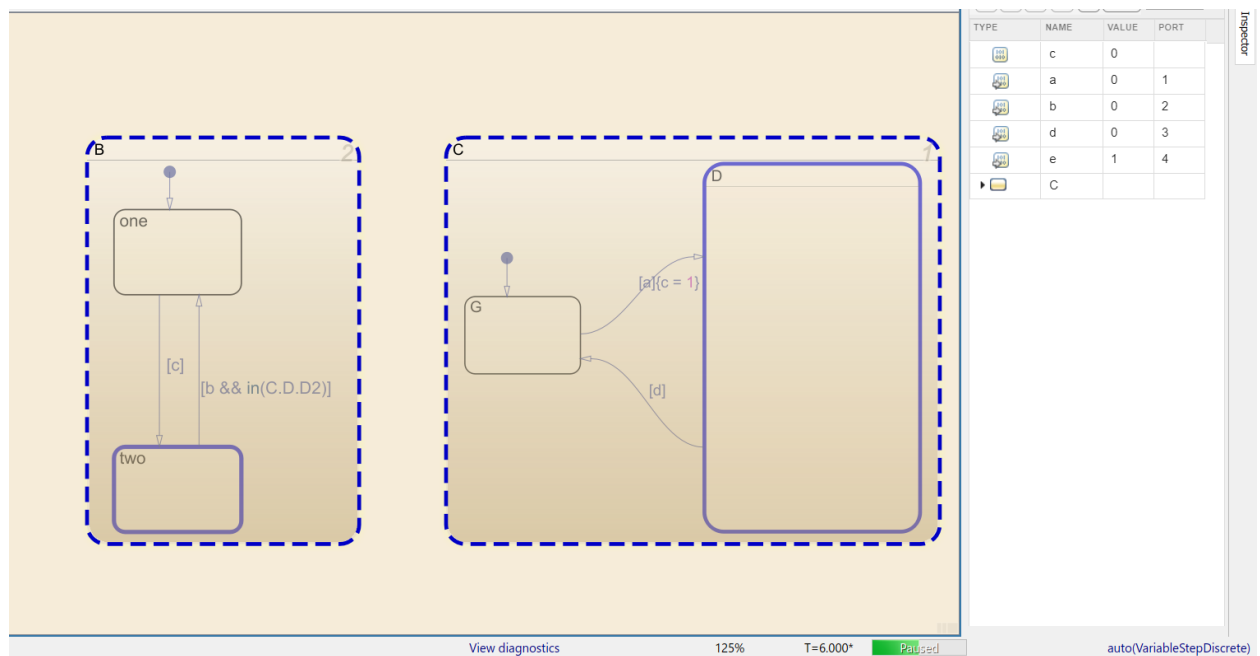
در زمان 2:



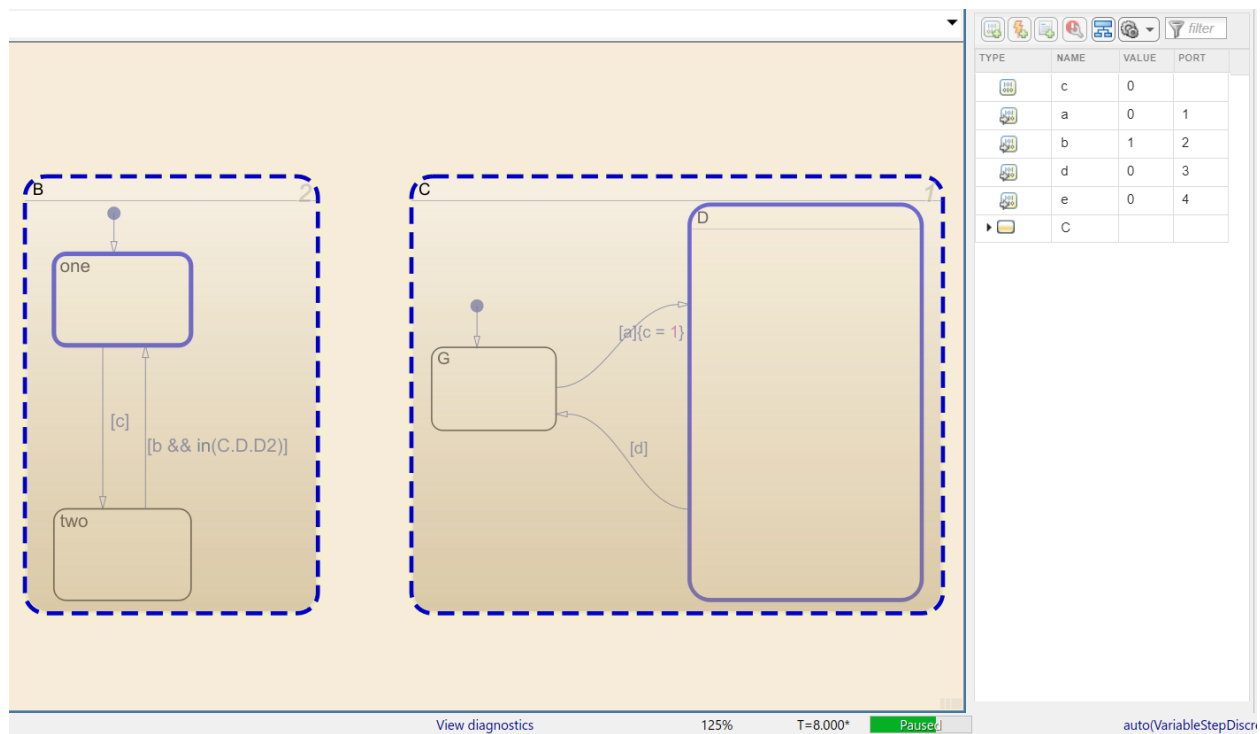
سپس در زمان 4 که مقدار $b=1$ شد:



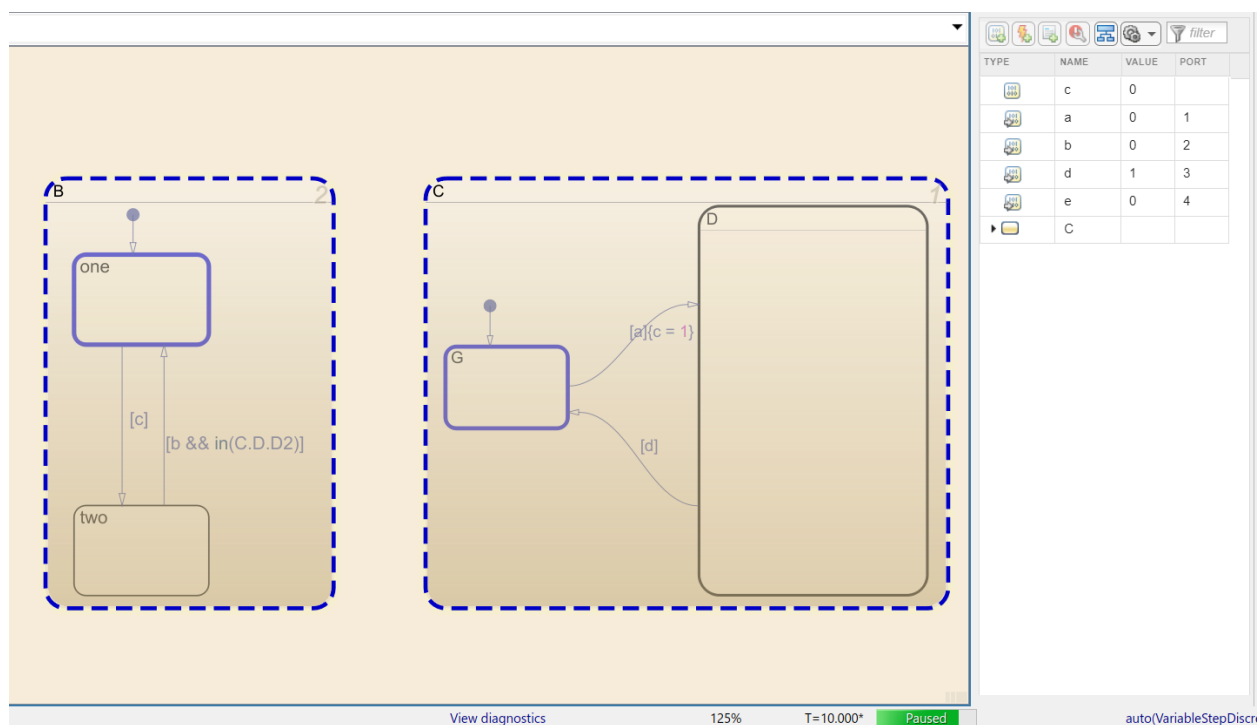
سپس در زمان 6 که مقدار $e=1$ شد:



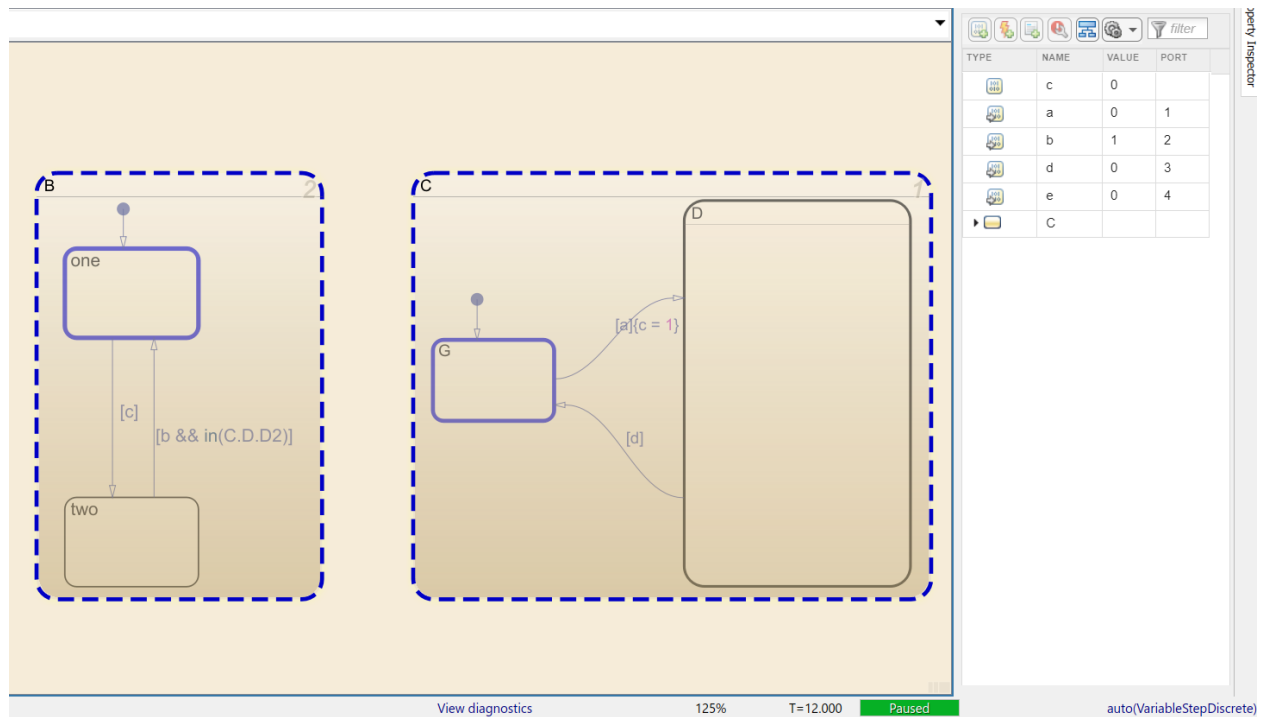
سپس در زمان 8 که مقدار $b=1$ شد دوباره:



سپس در زمان 10 که مقدار $d=1$ شد:



سپس در زمان 12 که مقدار $b=1$ شد:



سوال ۳

(الف)

با توجه به گراف sdf داده شده ، ماتریسی مینویسیم که سطرهای آن یال‌های گراف و ستون‌های آن راس‌های گراف است.

ماتریس توپولوژی:

-3 2 0 0

0 1 -3 0

0 0 4 -1

0 -4 0 3

مراحل بدست آوردن رنک ماتریس:

*ابتدا دترمینان ماتریس را محاسبه میکنیم و اگر مساوی 0 شد آنگاه یعنی دترمینان ماتریس کمتر از مینیمم سطر و ستون است.

** حال ماتریس را به ماتریس بالا مثلثی تبدیل میکنیم.

*** پس از تبدیل به ماتریس بالا مثلثی به قطر ماتریس نگاه کرده و به تعداد اعدادی که مخالف 0 هستند رنک ماتریس گوییم.

برای تبدیل به ماتریس بالا مثلثی:

$$4 * R_2 + R_4 \rightarrow R_4$$

$$3 * R_3 + R_4 \rightarrow R_4$$

و ماتریس به صورت زیر می شود:

$$\begin{matrix} -3 & 2 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 1 & -3 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 0 & 4 & -1 \end{matrix}$$

$$\begin{matrix} 0 & 0 & 0 & 0 \end{matrix}$$

که رنک ماتریس برابر ۳ است.

(ب)

معادلات را می نویسیم:

$$3a_0 = 2a_1$$

$$a_1 = 3a_2$$

$$4a_2 = a_3$$

$$3a_3 = 4a_1$$

حالا معادلات را بر حسب a_1 می نویسیم:

$$A_0 = 2/3 a_1$$

$$A2=1/3a1$$

$$A3=4/3a1$$

حال اگر $a1$ را ۳ در نظر بگیریم به این ترتیب باید fire داشته باشیم:

2

3

1

4

(ج)

روش اول)

ماتریس اول تعداد توکن موجود در بافر و ماتریس دوم تعداد دفعات مورد نیاز برای فایر کردن که در بخش ب بدست آوردیم است.

از $a0$ شروع کنیم:

$$\begin{array}{cccccccccccc} 4 & 1 & 3 & 3 & 3 & 3 & 0 & 2 & 2 & 4 & 4 \\ 2 & a_0 & 2 & a_1 & 3 & a_2 & 0 & a_3 & 0 & a_3 & 0 & a_0 & 0 & a_1 & 1 & a_3 & 1 & a_1 & 2 & a_3 & 2 \\ 0 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 4 & \rightarrow & 3 & \rightarrow & 2 & \rightarrow & 2 & \rightarrow & 2 & \rightarrow & 1 & \rightarrow & 1 & \rightarrow & 0 \\ 4 & 4 & 0 & 0 & 3 & 6 & 6 & 2 & 5 & 1 & 4 \end{array}$$

$$\begin{array}{cccccccccccc} 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & a_0 & 3 & a_1 & 2 & a_2 & 2 & a_3 & 2 & a_3 & 2 & a_0 & 2 & a_1 & 1 & a_3 & 1 & a_1 & 0 & a_3 & 0 \\ 1 & \rightarrow & 1 & \rightarrow & 1 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 0 \\ 4 & 4 & 4 & 4 & 3 & 2 & 2 & 2 & 1 & 1 & 0 \end{array}$$

پس زمانبندی تکرار شونده: $a0-a1-a2-a3-a3-a0-a1-a3-a1-a3$

روش دوم)

ابتدا از a_1 شروع کنیم:

0	2	2	2	2	4	1	1	3	0	0
2	a_1	3	a_2	0	a_3	0	a_3	1	a_0	1
0	\rightarrow	0	\rightarrow	4	\rightarrow	3	\rightarrow	2	\rightarrow	2
4	0	0	3	6	2	2	5	1	1	4

2	2	2	2	2	2	1	1	1	0	0
3	a_1	2	a_2	2	a_3	2	a_3	2	a_1	1
1	\rightarrow	1	\rightarrow	0	\rightarrow	0	\rightarrow	0	\rightarrow	0
4	4	4	3	2	2	2	1	1	1	0

پس زمانبندی تکرار شونده: $a_1-a_2-a_3-a_3-a_1-a_0-a_3-a_1-a_0-a_3$

مقدار بافر ها: $4+3+4+6=17$

(د)

منطق کد به این صورت است که در ابتدا بافر ها را آرایه های دوتایی تعریف میکنیم که اولین به ترتیب نشان دهنده ی مقدار توکن تولیدی و مصرفی هر یال در هر firing میباشد.

ابتدا ۴ آرایه به عنوان بافر برای ۴ یال تعریف کردیم و initial token ها را در بخش مصرف قرار دادیم.

```
1
2  #include <iostream>
3
4  using namespace std;
5  //arrays={consume , produce}
6  int a0a1[] = {4,0};
7  int a1a2[] = {2,0};
8  int a2a3[] = {0,0};
9  int a3a1[] = {4,0};
```

سپس داخل توابع a0 تا a3 را پیاده سازی کردیم.

```
11 void a0(){
12     a0a1[0]-=3;
13 }
14 void a1(){
15     a1a2[0]+=1;
16     a0a1[1]+=2;
17     a3a1[0]-=4;
18 }
19 void a2(){
20     a1a2[0]-=3;
21     a2a3[1]+=4;
22 }
23 void a3(){
24     a2a3[0]-=1;
25     a3a1[1]+=3;
26 }
```

```

27 void typeBuffer(){
28     for (int i = 0; i < 2; ++i) {
29         cout<<" a0a1["<<i<<"]="<<a0a1[i];
30     }
31     cout<<"\n";
32     for (int i = 0; i < 2; ++i) {
33         cout<<" a1a2["<<i<<"]="<<a1a2[i];
34     }
35     cout<<"\n";
36     for (int i = 0; i < 2; ++i) {
37         cout<<" a2a3["<<i<<"]="<<a2a3[i];
38     }
39     cout<<"\n";
40     for (int i = 0; i < 2; ++i) {
41         cout<<" a3a1["<<i<<"]="<<a3a1[i];
42     }
43     cout<<"\n";
44 }

```

سپس در این بخش از کد بالا پرینت های بافر هارا انجام دادیم.

```
45  int main(){
46      a0();
47      cout<<"a0:"<<"\n";
48      typeBuffer();
49      a1();
50      cout<<"a1:"<<"\n";
51      typeBuffer();
52      a2();
53      cout<<"a2:"<<"\n";
54      typeBuffer();
55      a3();
56      cout<<"a3:"<<"\n";
57      typeBuffer();
58      a3();
59      cout<<"a3:"<<"\n";
60      typeBuffer();
61      a0();
62      cout<<"a0:"<<"\n";
63      typeBuffer();
64      a1();
65      cout<<"a1:"<<"\n";
66      typeBuffer();
67      a3();
68      cout<<"a3:"<<"\n";
69      typeBuffer();
70      a1();
71      cout<<"a1:"<<"\n";
72      typeBuffer();
73      a3();
74      cout<<"a3:"<<"\n";
75      typeBuffer();
76  }
```

در آخر زمانبندی را گذاشتیم.