**به نام خدا**

تمرین دوم درس سیستم‌های نهفته بی‌درنگ

نیمسال دوم 1401-1402

**سوال2)**

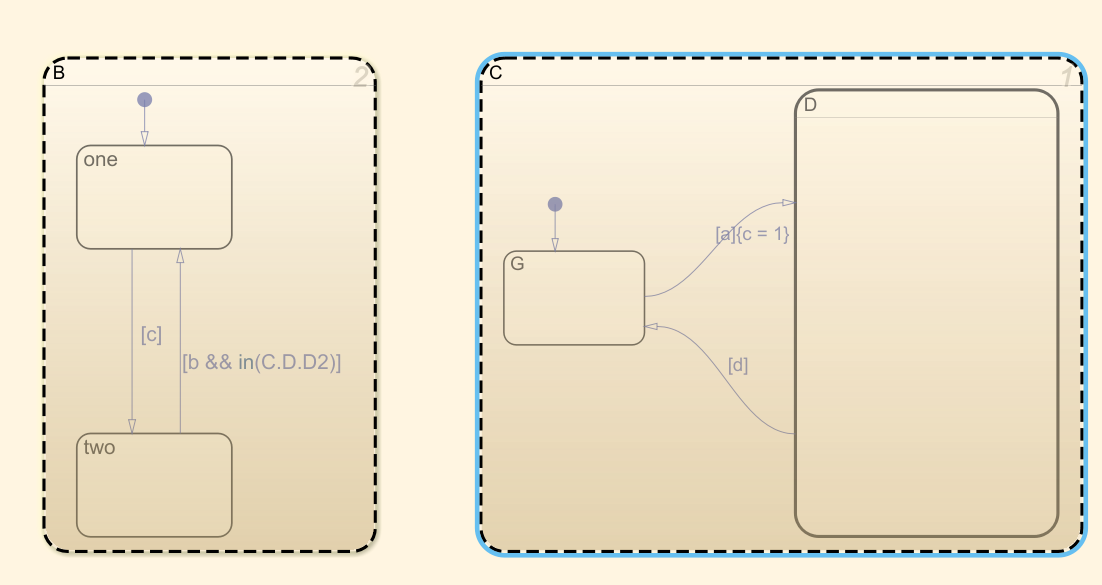
الف)

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.

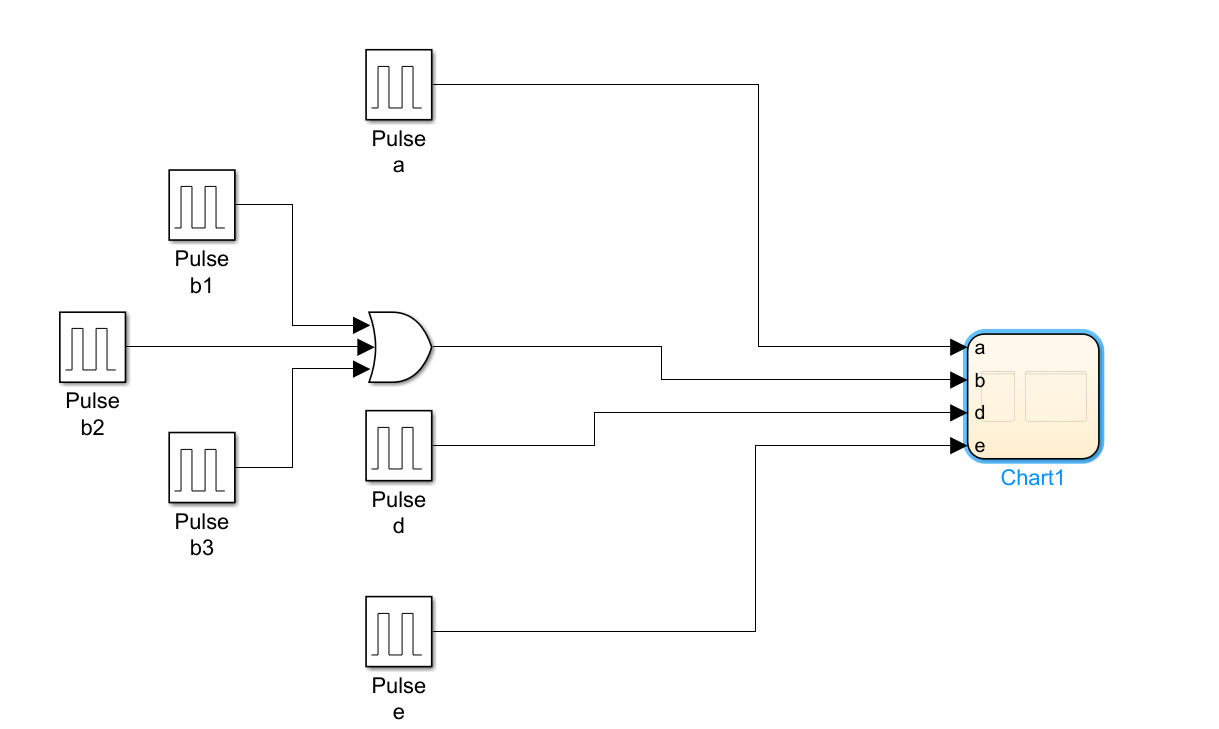
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **D2** | **D1** | **D** | **G** | **2** | **1** | **C** | **B** |
|  |  |  | **x** |  | **x** | **x** | **x** | **Init** |
|  | **x** | **x** |  | **x** |  | **x** | **x** | **a** |
|  | **x** | **x** |  | **x** |  | **x**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

ب)

ابتدا استیت چارت را در سیمولینک رسم میکنیم:



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



5 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:

1-ابتدا ورودی a با تاخیر زمانی 2 ثانیه.

2-ورودی b با تاخیر زمانی 4ثانیه

3-ورودیe با تاخیر زمانی 6ثانیه.

4-ورودی b با تاخیر زمانی 8 ثانیه

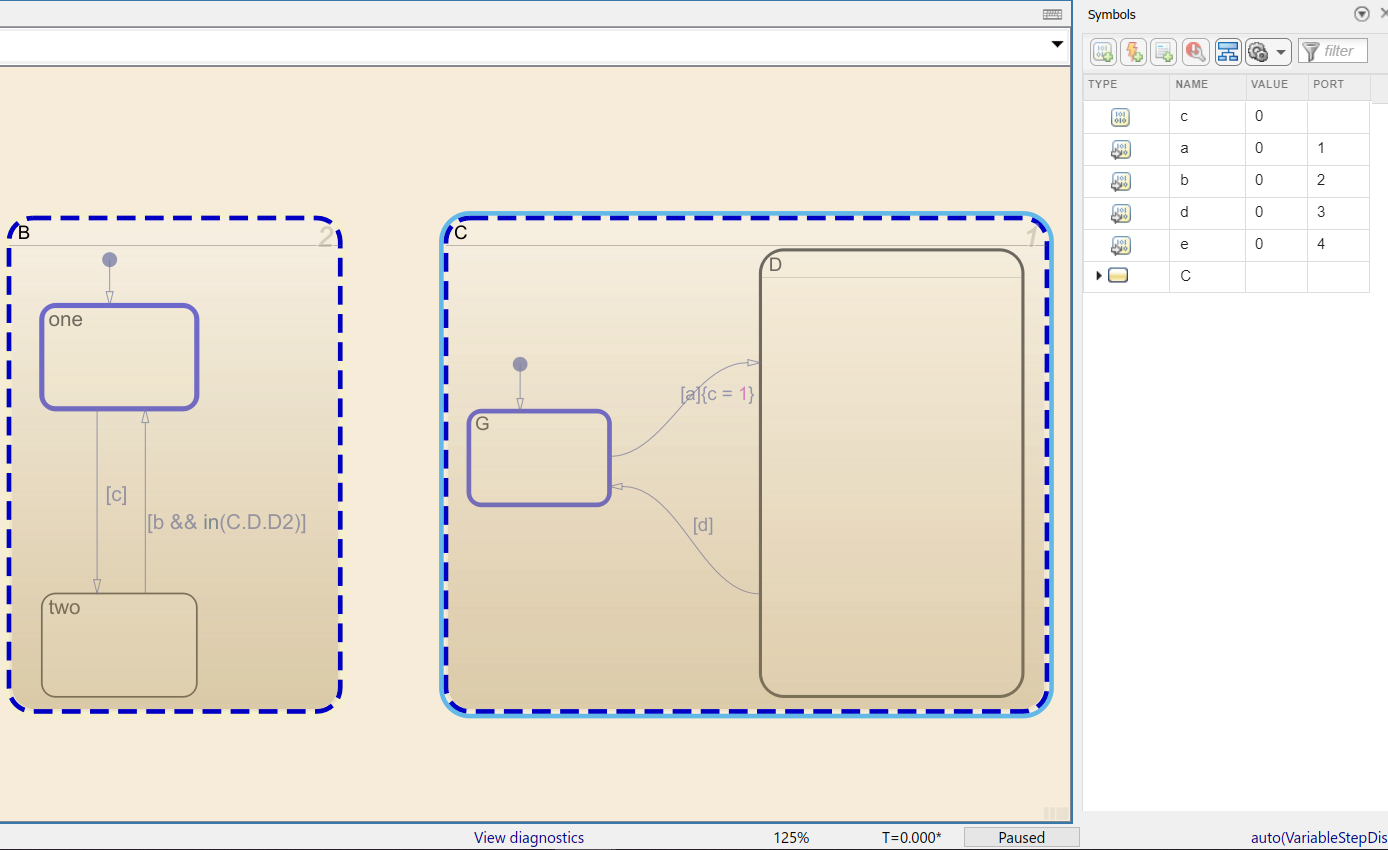
5- ورودی d با تاخیر زمانی 10 ثانیه

6-ورودی b با تاخیر زمانی 12 ثانیه.

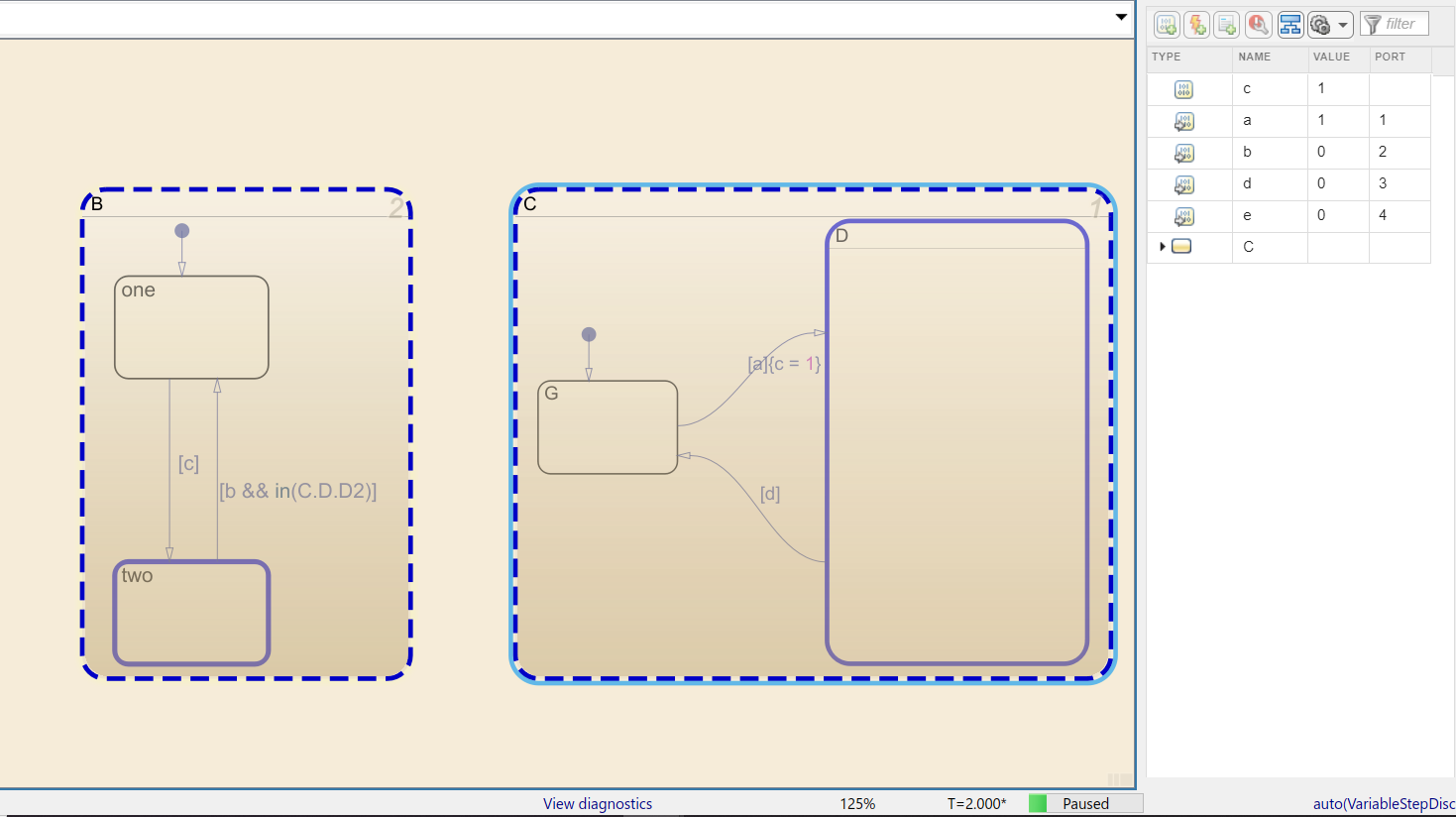
سپس در view symbol را انتخاب کرده و مقدار دهی می کنیم.

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

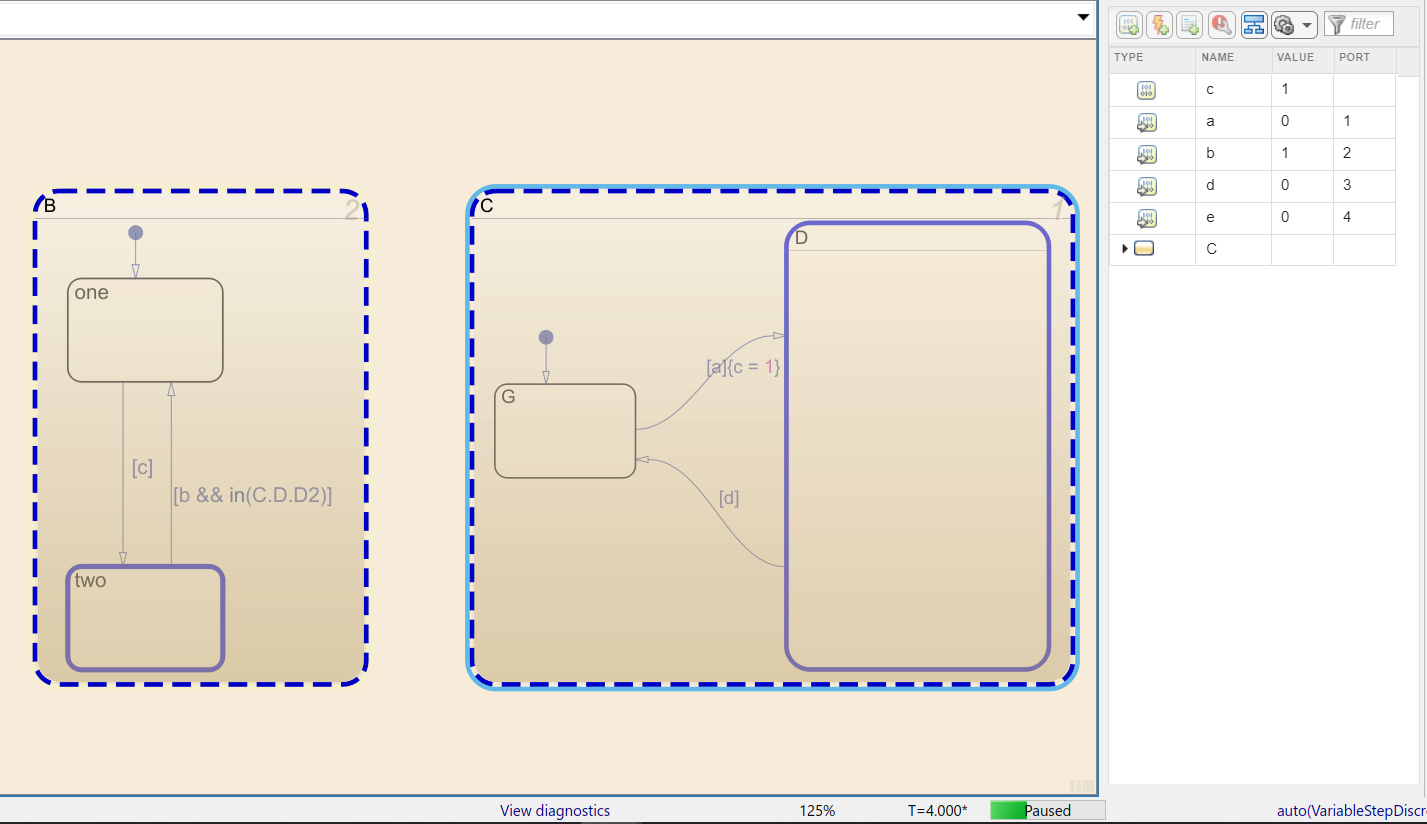
ابتدا در زمان 0 هستیم:



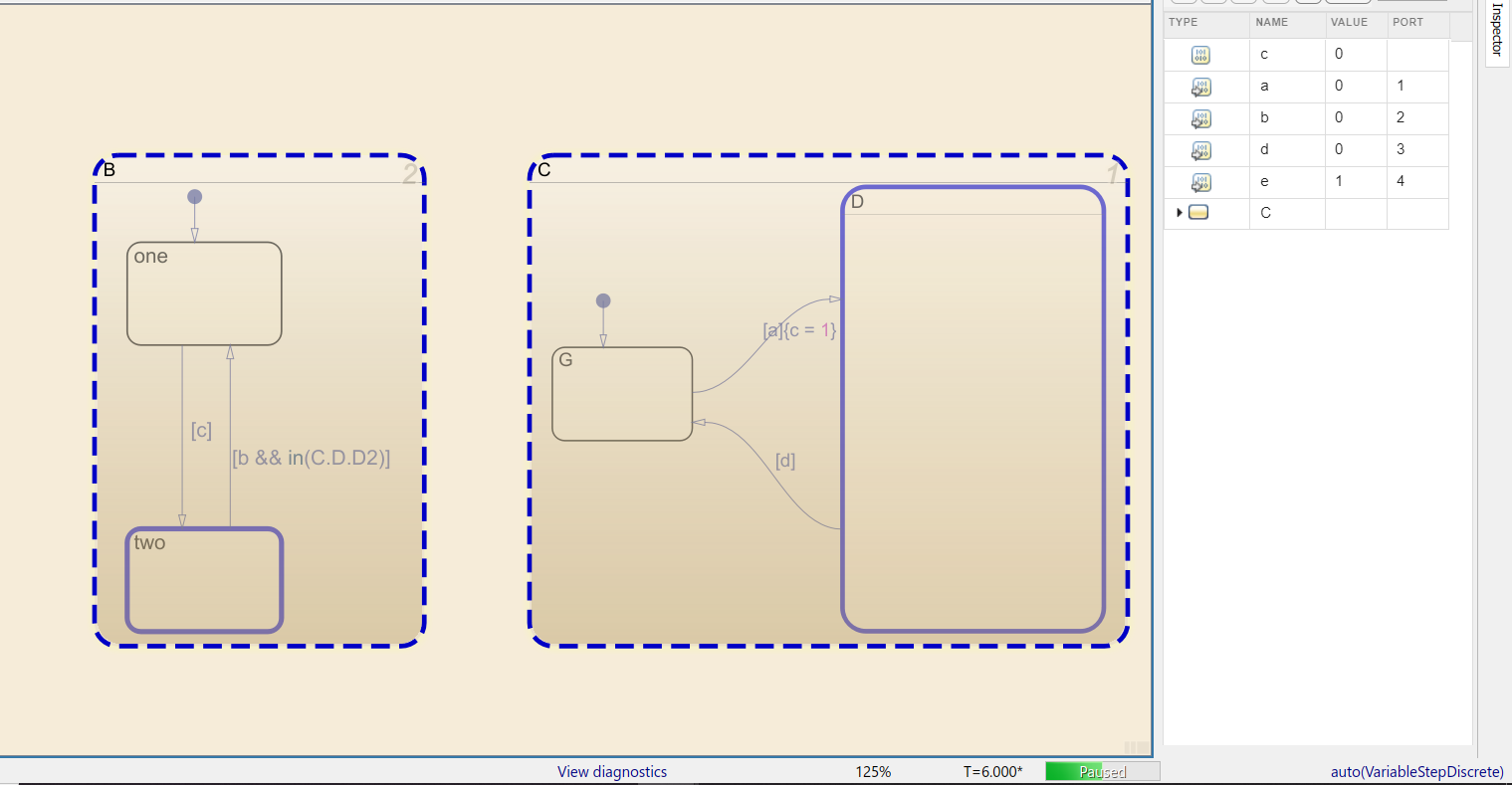
در زمان 2 :



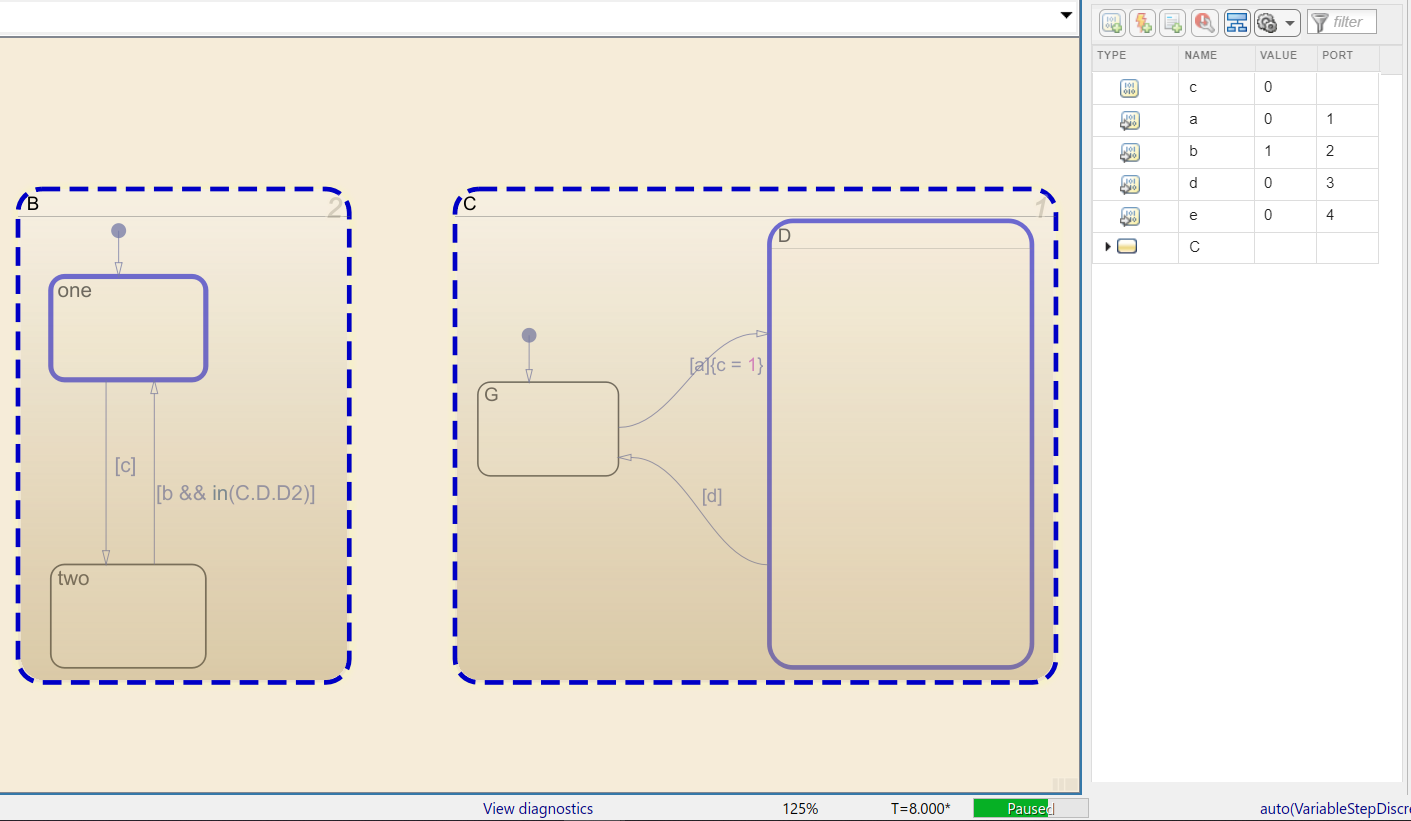
سپس در زمان4 که مقدار b=1 شد:



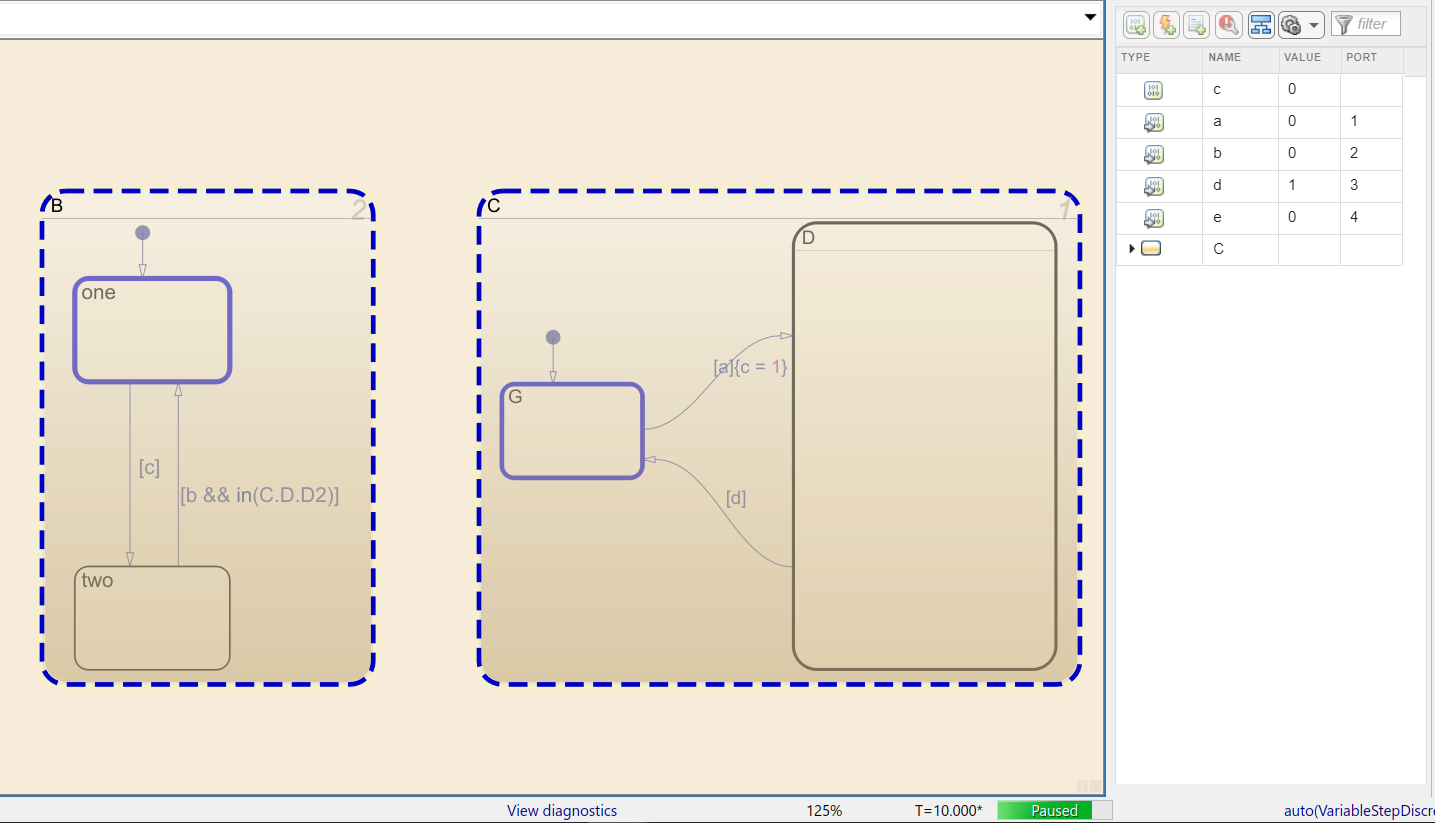
سپس در زمان 6 که مقدار e=1 شد:



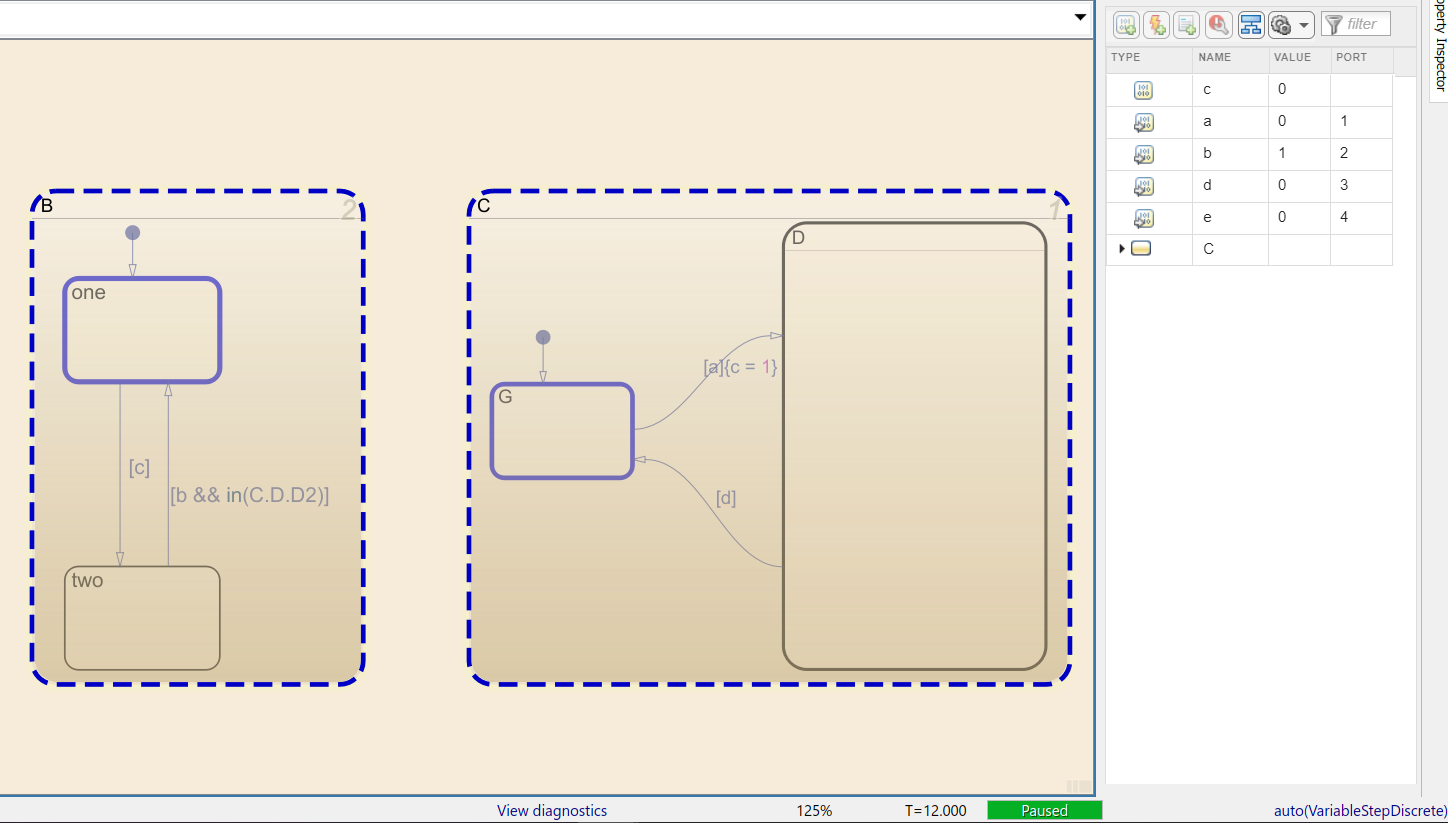
سپس در زمان 8 که مقدار b=1 شد دوباره:



سپس در زمان10 که مقدار d=1 شد:



سپس در زمان 12 که مقدار b=1 شد:



**سوال3)**

الف)

با توجه به گراف sdf داده شده ، ماتریسی مینویسیم که سطرهای آن یال‌های گراف و ستون های آن راس های گراف است.

ماتریس توپولوژی:

-3 2 0 0

0 1 -3 0

0 0 4 -1

0 -4 0 3

مراحل بدست اوردن 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 مارتیس 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\*R2+R4🡪R4

3\*R3+R4🡪R4

و ماتریس به صورت زیر می شود:

-3 2 0 0

0 1 -3 0

0 0 4 -1

0 0 0 0

که رنک ماتریس برابر 3 است.

ب)

معادلات را می نویسیم:

3a0=2a1

a1=3a2

4a2=a3

3a3=4a1

حالا معادلات را برحسب a1 می نویسیم:

A0=2/3a1

A2=1/3a1

A3=4/3a1

حال اگر a1 را 3 درنظر 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 اوردیم است.

از a0 شروع کنیم:

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

روش دوم)

ابتدا از a1 شروع کنیم:

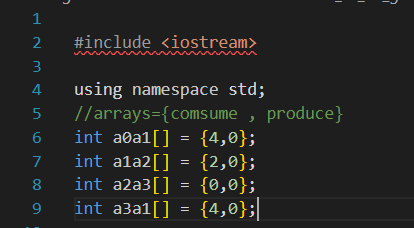
پس زمانبندی تکرار شونده: a1-a2-a3-a3-a1-a0-a3-a1-a0-a3

مقدار بافر ها: 4+3+4+6=17

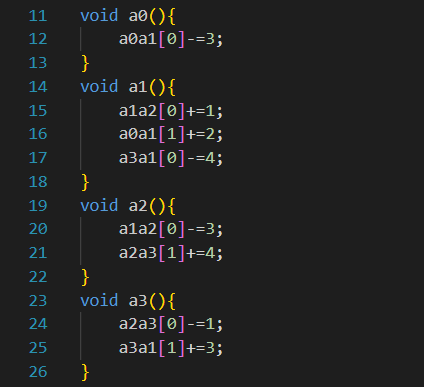
د)

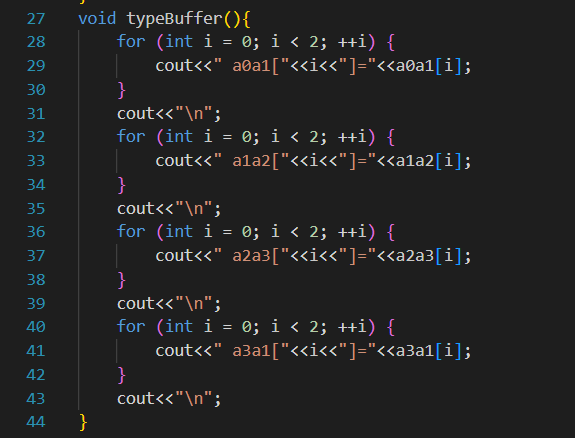
منطق کد به این صورت است که در ابتدا بافر ها را آرایه های دوتایی تعریف میکنیم که اولین به ترتیب نشان دهنده ی مقدار توکن تولیدی و مصرفی هر یال در هر firing میباشد.

ابتدا 4 ارایه به عنوان بافر برای 4 یال تعریف کردیم و initial token ها را در بخش مصرف قرار دادیم.

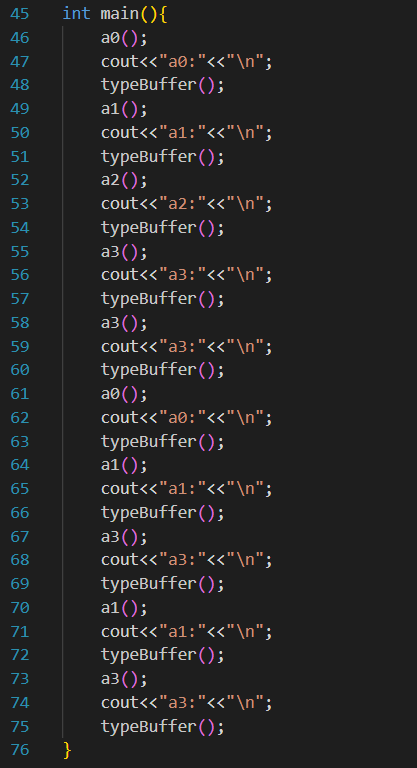


سپس داخل توابع a0تا a3 را پیاده سازی کردیم.





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



در آخر زمانبندی را گذاشتیم.