

- در ابتدا asm chart و CU و DP مدار خواسته شده را میکشیم و کامپوننت ها و سیگنال های کنترلی را مشخص میکنیم .  
(پیوست شده)
- نیاز به ذکر است که asm ما دارای یک حلقه دو بلاکه است که این حلقه 7 بار میچرخد پس تعداد کلاک مورد نیاز برای این asm به این صورت است  $1 + 1 + 1 + 1 + 7 * 2 + 1 + 1$   
برای توضیح بیشتر :

```

1 : start_1
1 : start_2
1 : init
7 * 2 : (cal1 and cal2)
1 : cal1
1 : end

```

- برای پیاده سازی این کد ، از کامپوننت های کشیده شده در قسمت اول استفاده میکنیم ، بر روی حافظه rom هم باید مقادیر سازگار با بسط تیلور را هارد کد کنیم .  
نکات تکمیلی به طور کامنت بر روی کد آمده است.
- برای قسمت پیاده سازی پایتون ، همانند الگوریتم داده شده در صورت سوال عمل کردیم. (فایل جوینتر پیوست شده)
- حال برای مقایسه real-time بودن ، با چند بار ران کردن کد جوینتر به این نتیجه میرسیم که زمان ران شدن این الگوریتم در پایتون consistent نیست ، اما در FPGA ای که داریم اوضاع بدین گونه نیست و در هر مرحله ران کردن ، ما جواب را بعد از 19 کلاک به دست میاوریم ، به همین دلیل FPGA داده شده از لحاظ real-time بهتر است و در سیستم های نهفته استفاده میشوند.  
حال برای مقایسه عددی :

Time in python :  $5 * 10^{-6}$

Time in FPGA :  $19 \text{ clock} * (1/200\text{Mhz}) \text{ time} / \text{clock} = 10^{-7}$

- در اخر لازم به ذکر است که در کد وریلاگ تلاش شده که مطابق اطلاعات بالا برنامه پیاده سازی شود .