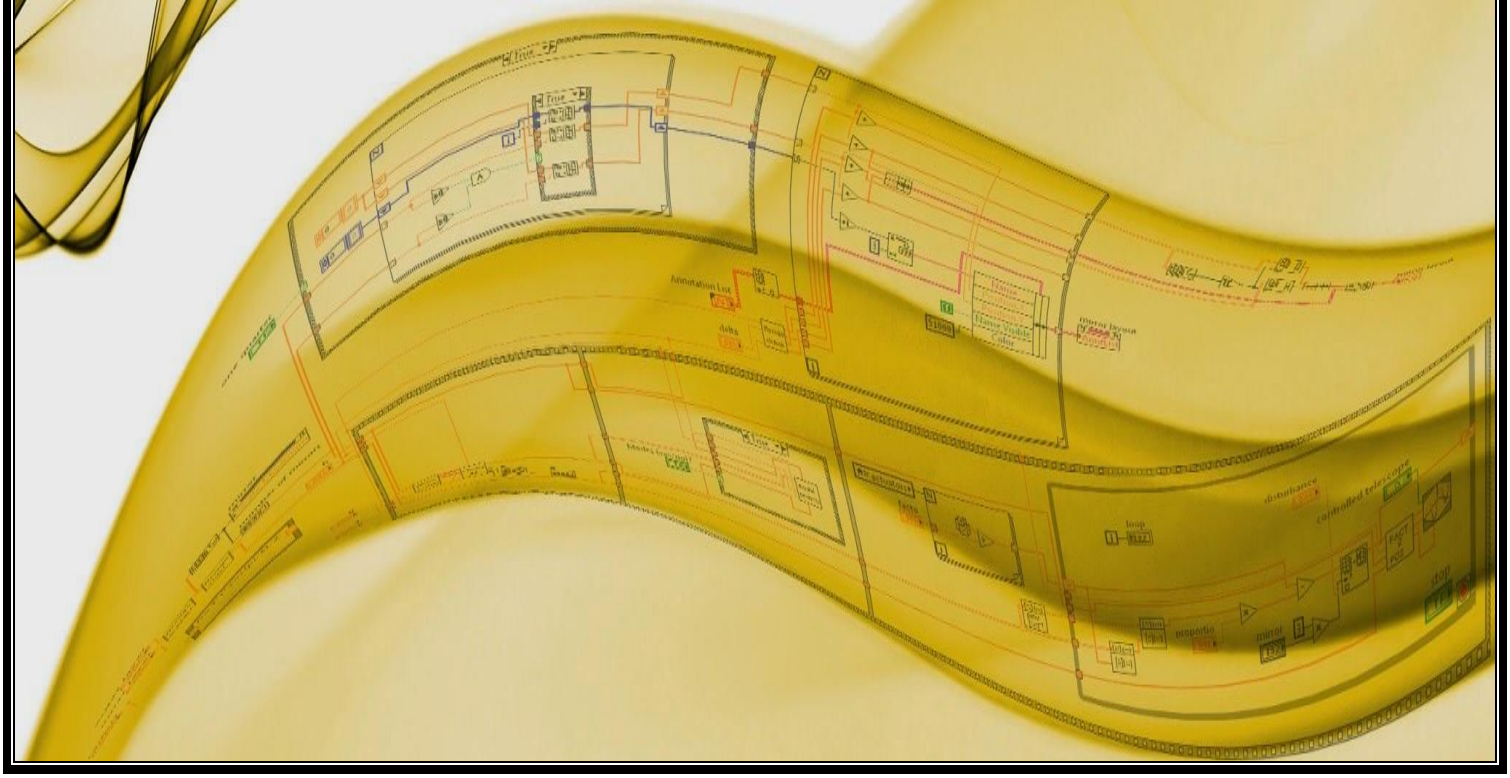
Programing with Labview

ابراهیم معظمی

میکرودیزاینرالکترونیک

Melec.ir



فصل اول: آشنایی با محیط labview

فصل دوم : ساختار ها و حلقه ها در labview

فصل سوم: آرایه ها در labview

فصل چهارم: نمودار ها و گراف ها در labview

فصل پنجم: رشته ها در labview

فصل ششم: ریاضیات در labview

فصل هفتم: توابع پیشرفته موجود در labview

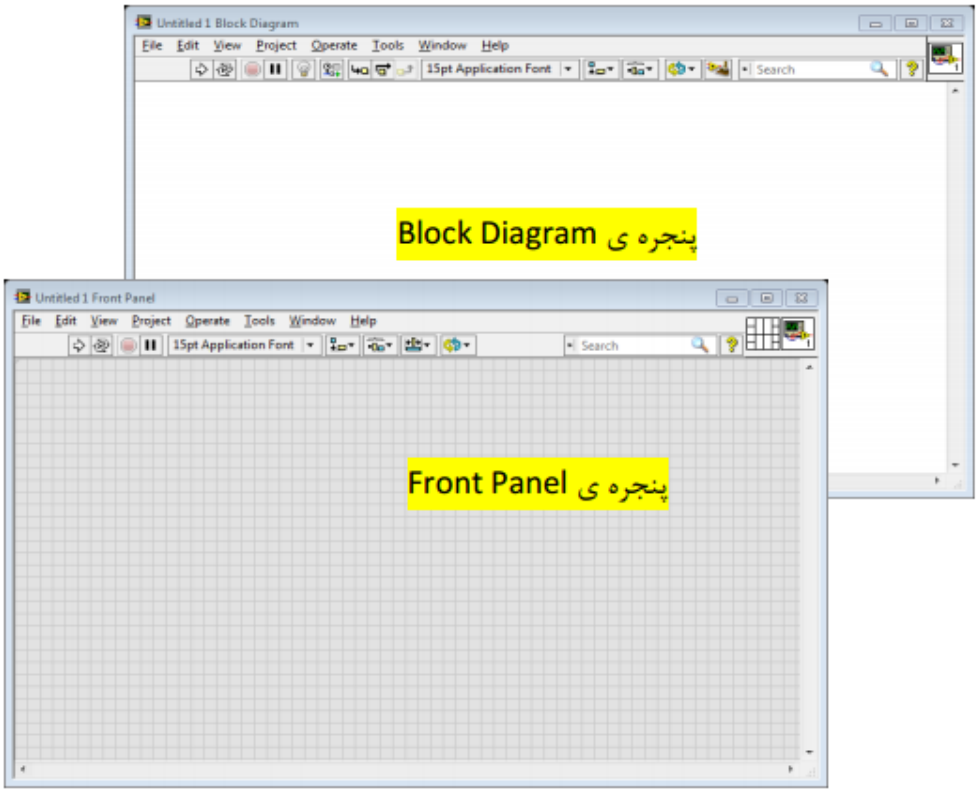
فصل هشتم : آموزش تولکیت کنترل نرم افزار labview

فصل نهم : ارتباط سریال با labview

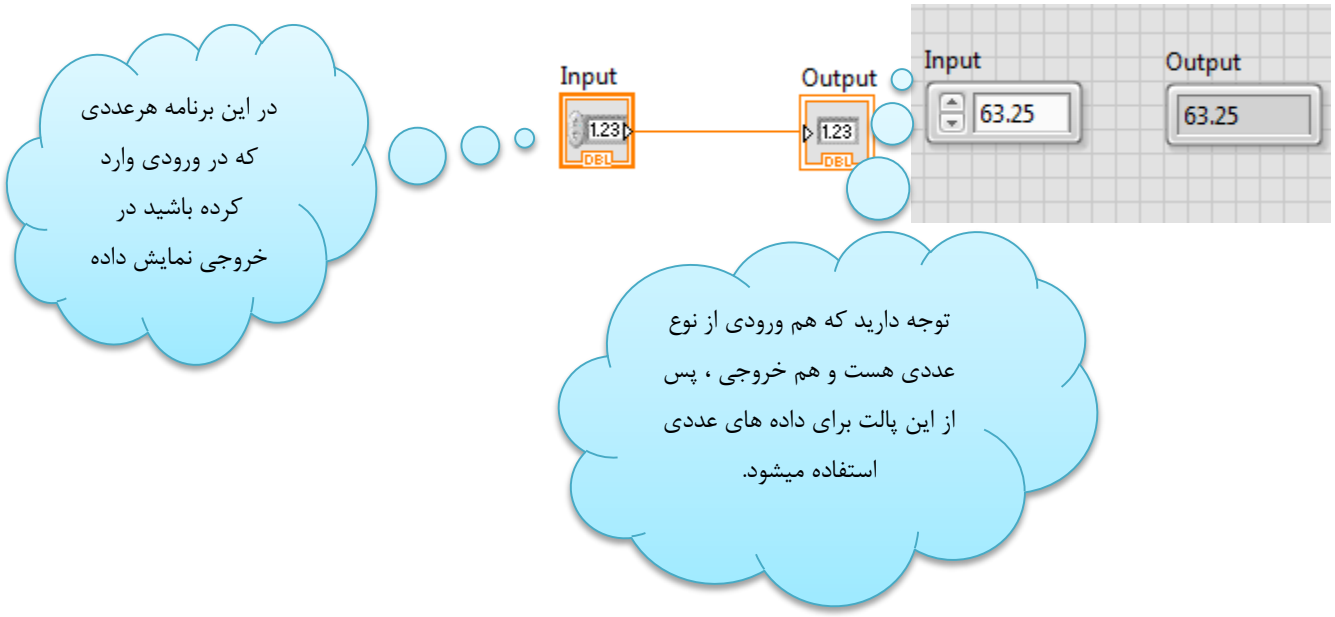
فصل دهم : ارتباط شبکه با labview

فصل یازدهم : برنامه نویسی میکروکنترلر ARM با نرم افزار labview

فصل دوازدهم: برنامه نویسی FPGA با نرم افزار

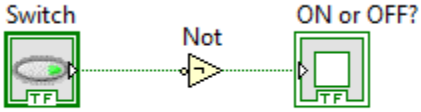


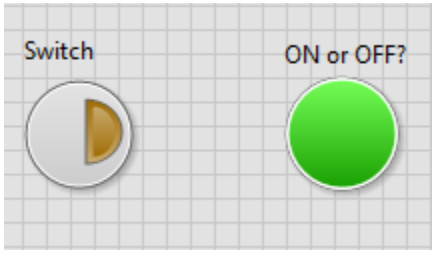
مثال : برنامه ای بنویسید که در ورودی هرچه وارد میکنید در خروجی نمایش داده شود .



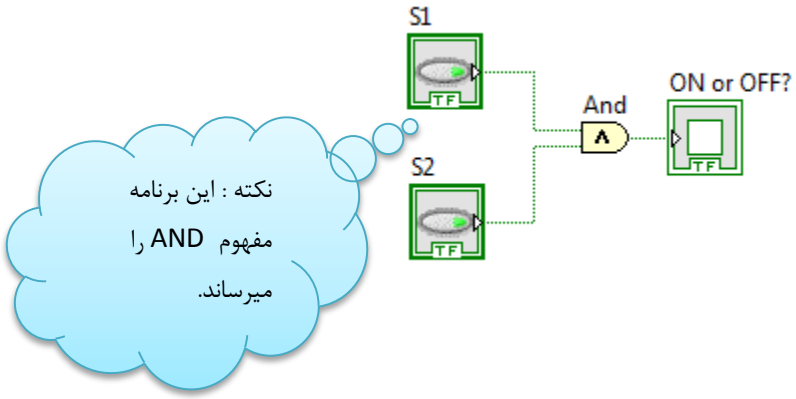
مثال : برنامه ای بنویسید که اگر سویچ ورودی روشن بود یک led روشن شود و اگرسویچ ورودی خاموش بود led خاموش شود.

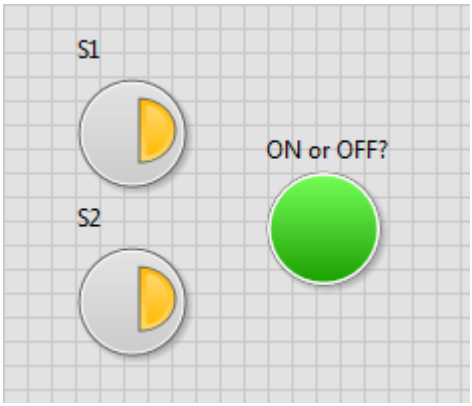
مثال : برنامه ای بنویسید که عکس مثال قبل عمل کند .





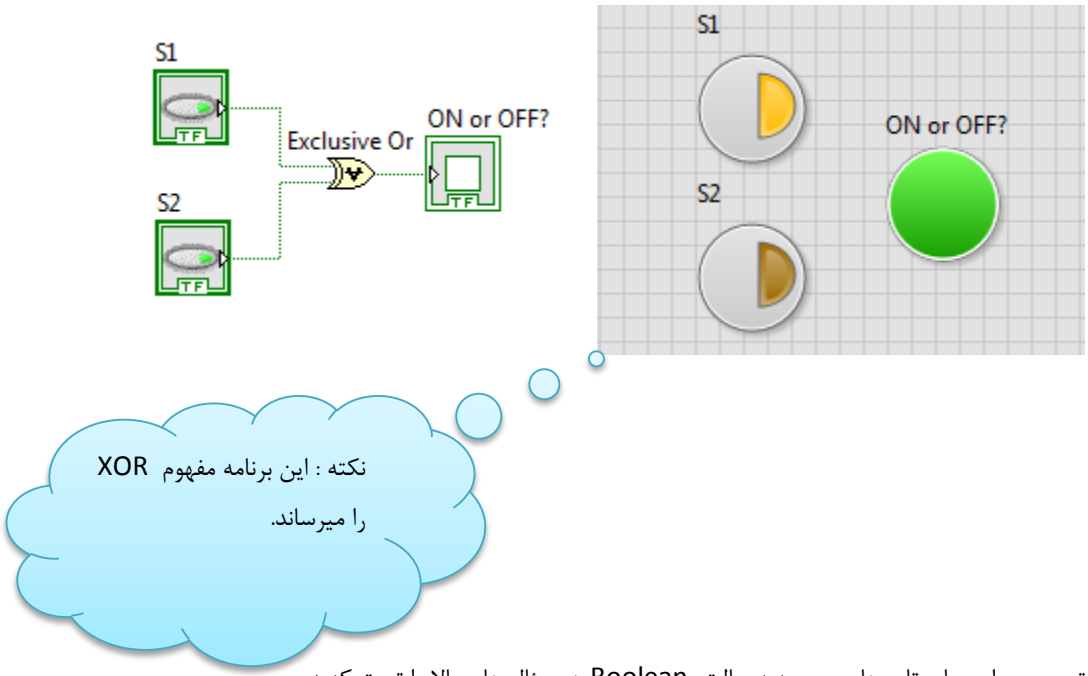
مثال : برنامه ای بنویسید که اگر دو تا ورودی یک بودند آن گاه یک led روشن شود در غیر این صورت led خاموش باشد .





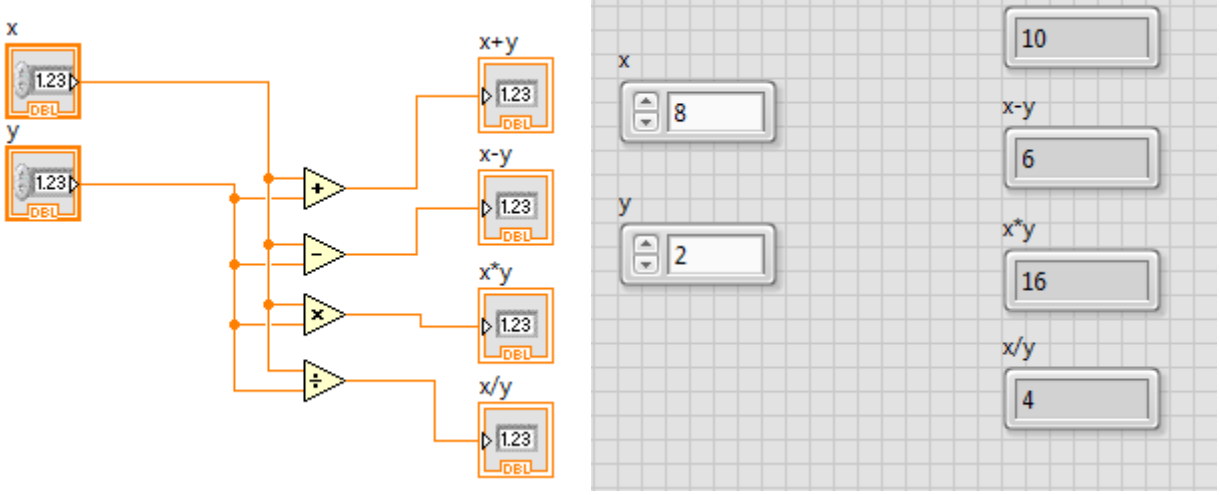
مثال : برنامه ای بنویسید که اگر هردو کلید ورودی خاموش بودند آنگاه led خروجی خاموش بشود.

مثال : برنامه ای بنویسید که اگر دو ورودی مثل هم نبودند led در خروجی روشن بشود.

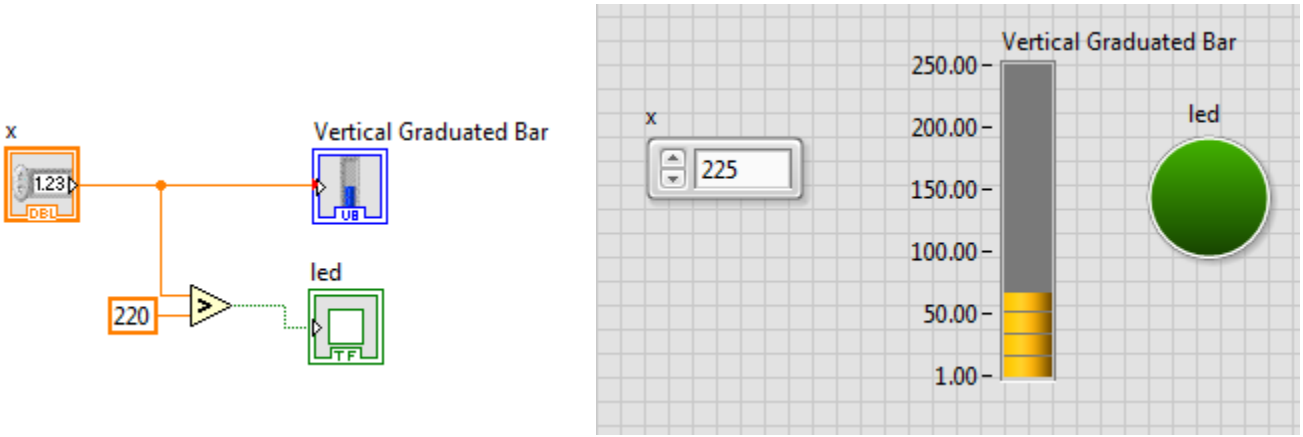


تمرین : برای سایر تابع های موجود در پالت Boolean هم مثال های بالا را تست کنید.

مثال : برنامه ای بنویسید که کامپیوتر دو تا عدد از شما بگیرد و حاصل ضرب و حاصل جمع و حاصل تقسیم و حاصل تفرق این دو عدد رو برای شما نشان بدهد.

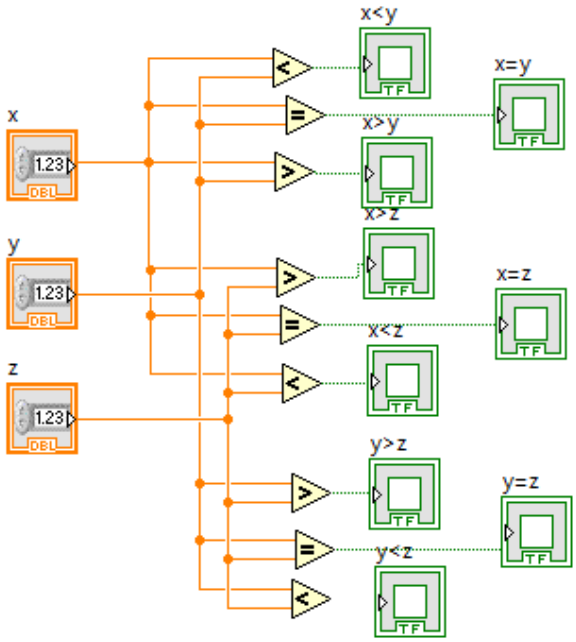


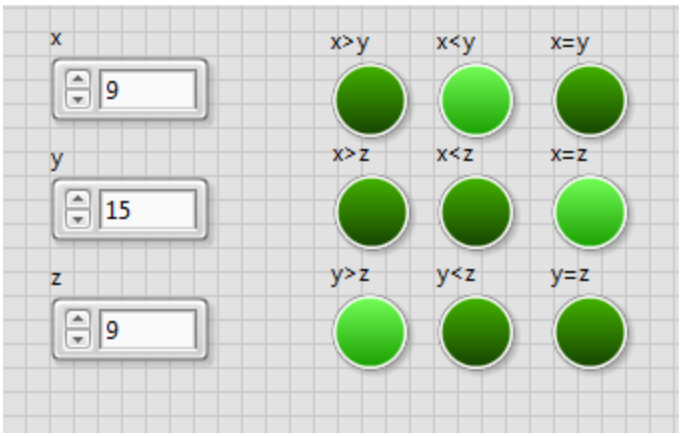
مثال : برنامه ای بنویسید که اگر ورودی از 220 بیشتر شد در خروجی یک led روشن شود.



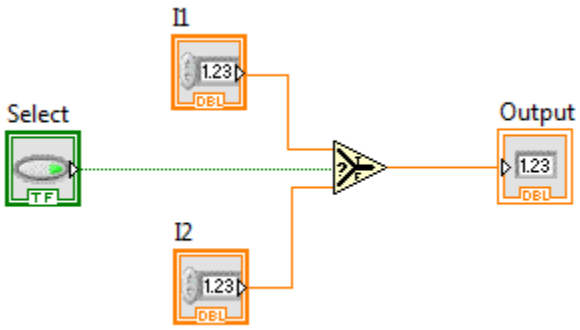
توجه : شاید به نظر این مثال ساده برسد اما خالی از ایده ی طراحی نیست شما میدانید برای هر قطعه ای ما باید پروتکشن (محافظت) انجام بدیم. در عمل به این کار میگویند بستن مدار گارد ، این برنامه میتواند بخشی از یک برنامه ی عملی باشد. مثلا فکر کنید قرار است یک ولتاژی به یک موتوری اعمال شود. از این روی بایستی در ابتدا چک شود که ولتاژ کوچک تر از آستانه ی تحمل موتور فرارتر نرود .

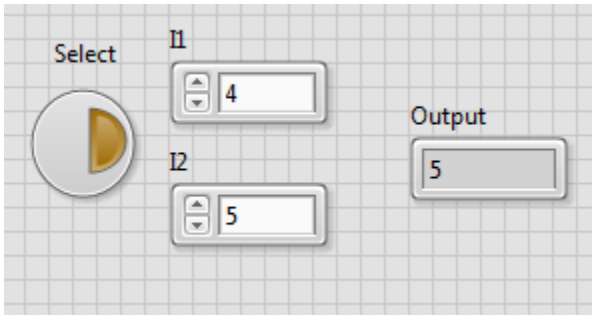
مثال : برنامه ای بنویسید که سه عدد ورودی را بگیرد x, y,z اگر y بزرگ تر از x بود یا کو چکتر و یا مساوی آنگاه چراغی روشن که معلوم شود y>x و یا y<x و یا x=y است . و به همین ترتیب برای z,y تکرار کنید.





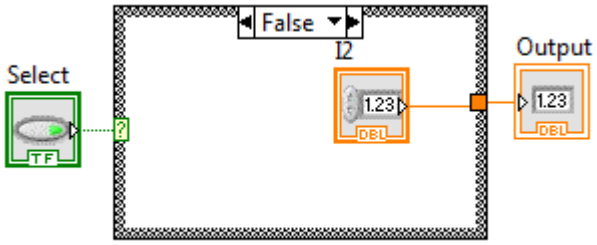
مثال : برنامه ای بنویسید که اگر کلید ورودی روشن بود ورودی 1را به خروجی ببرد اگر نه ورودی 2را به خروجی ببرد.

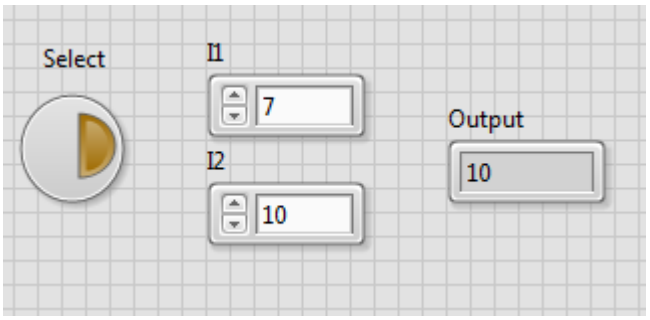




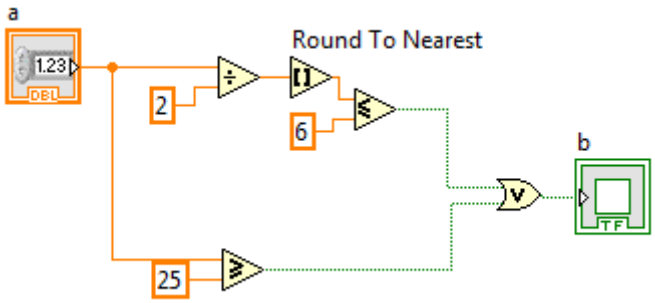
مثال : مثال بالا را به یک روش دیگر بنویسید .

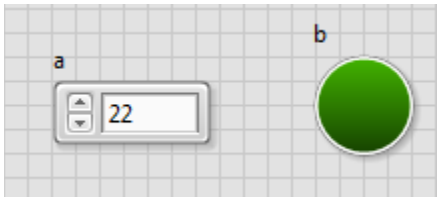
البته فعلا حلقه ی case را نگفته ام ولی بدانید که به روش زیر هم میتوان این برنامه را نوشت :



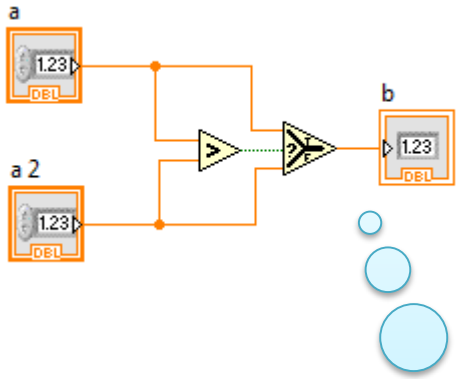


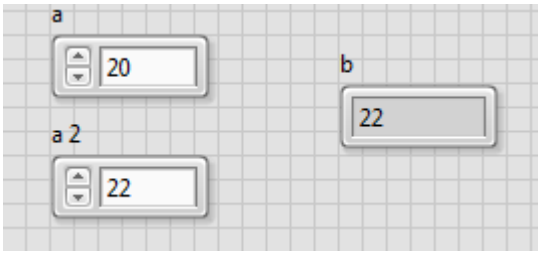
مثال : برنامه ی زیر را بررسی کنید :





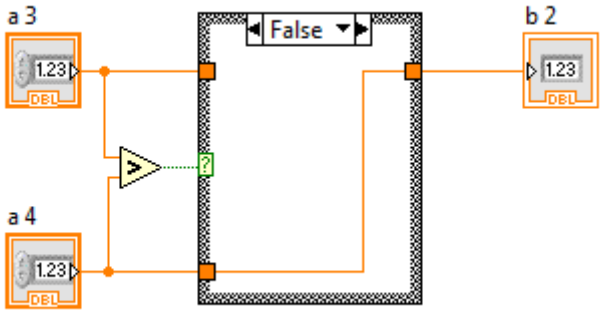
مثال : برنامه ی زیر را بررسی کنید :

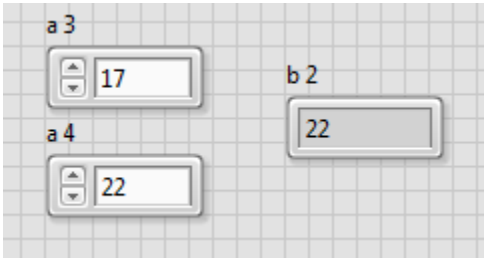




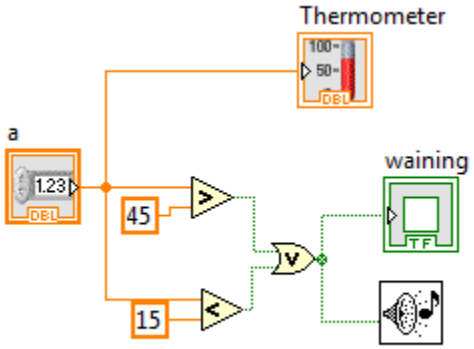
توضیح : چنانچه ورودی اول از ورودی دوم بزرگتر باشد آن گاه ورودی اول به خروجی منتقل میشود. واگر ورودی اول از ورودی دوم کوچکتر باشد. ورودی دوم به خروجی منتقل خواهد شد.

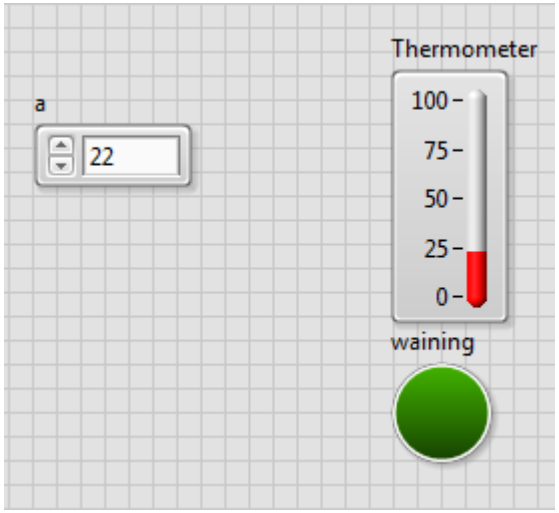
مثال : برنامه ی بالا را به روش دیگری بنویسید.





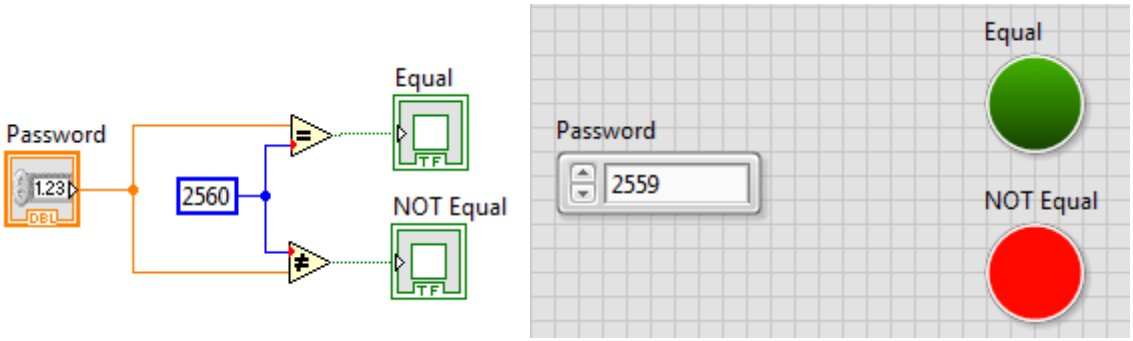
مثال : برنامه ی زیر را بررسی کنید و بگویید که در عمل میتواند برنامه ی چه وسیله ای باشد؟



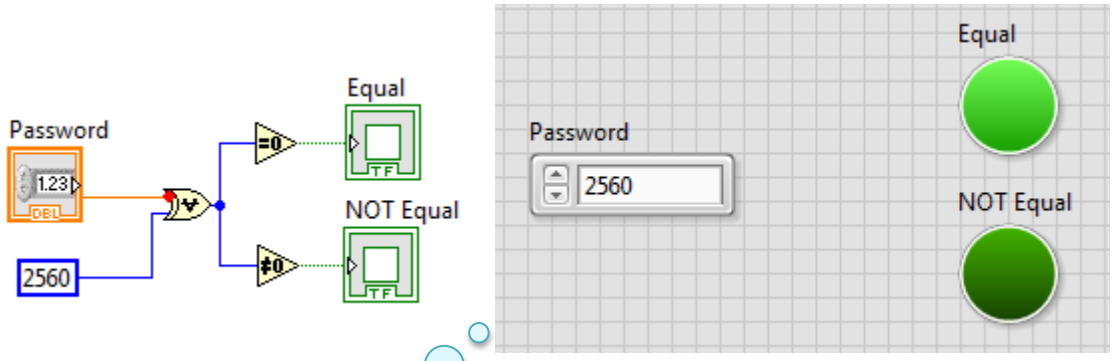


این برنامه میتواند بخشی از برنامه ی یک دماسنج باشد. که اگر دما بیشتر از 45 درجه شد و یا کمتر از 15 درجه شد هشدار بدهد.

مثال : برنامه ای بنویسید که از شما یک رمز در یافت کند و با رمز اصلی مقایسه کند اگر رمز وارد شده توسط شما درست بود آنگاه چراغ سبز روشن شود و اگر رمز را اشتباه وارد کردید چراغ قرمز روشن شود



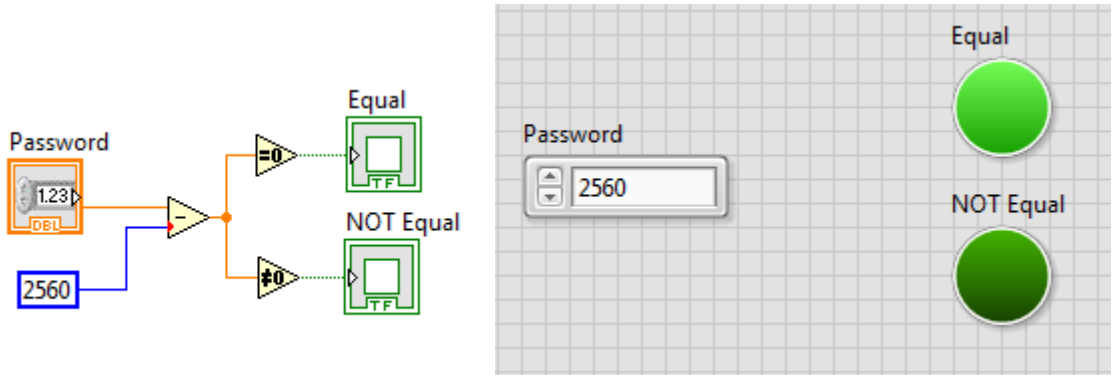
مثال : برنامه ی قبل را با یک تابع XOR بنویسید :



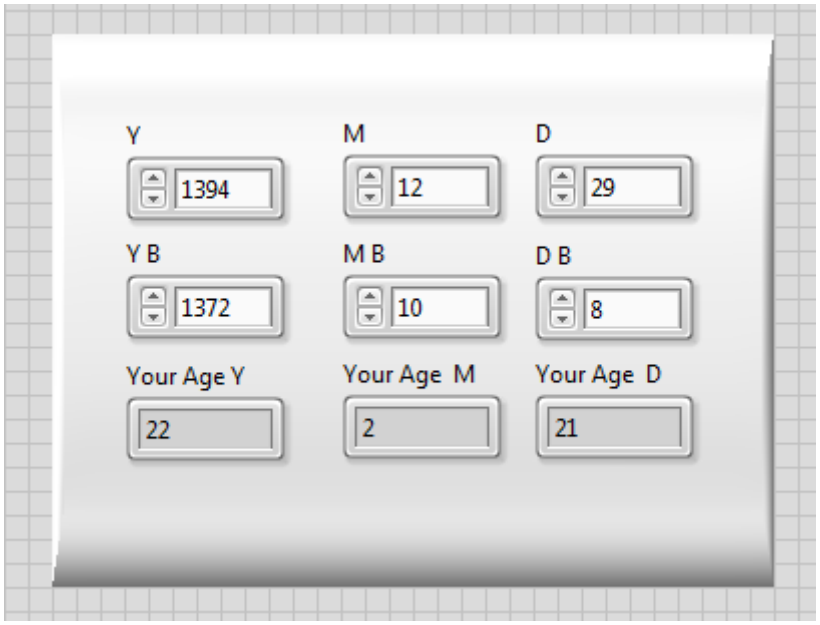
توجه : هر عددی با خودش

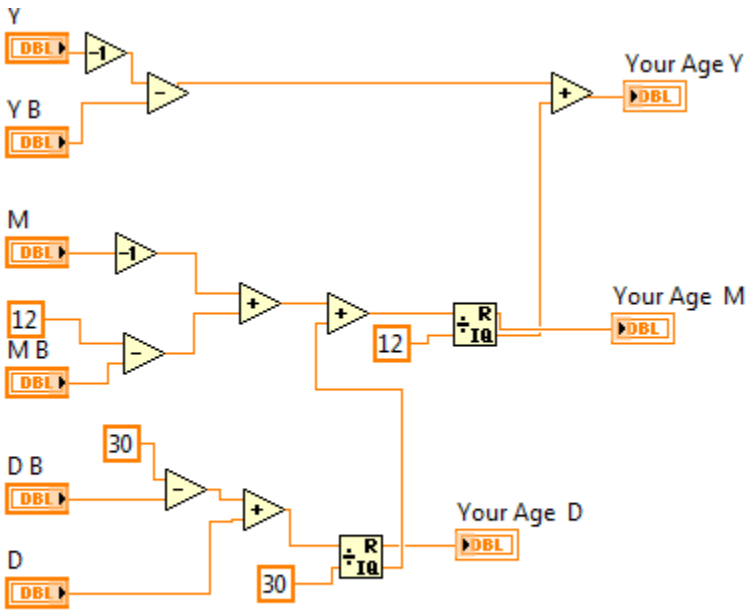
XORبشود نتیجه صفر خواهد شد .

مثال : برنامه ی بالا را با یک منها بنویسید :

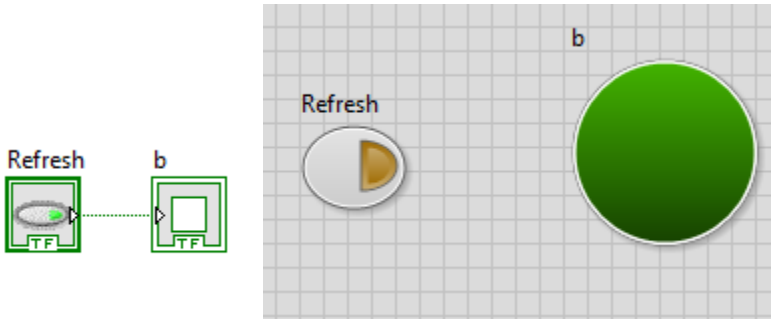


مثال : برنامه ای بنویسید که سال و ماه و روز تولد شما و سال و ماه و روز فعلی را بگیرد و سن شما را به سال و ماه و روز حساب کند .(فرض کند . (فرض : هر سال 21ماه و هرماه 03روز و هر روز 12ساعت است.





مثال : به برنامه ی زیر نگاه کنید :



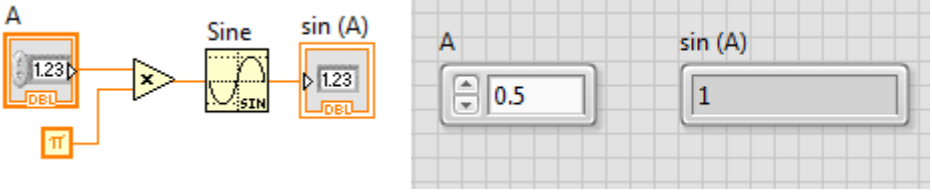
اگر مثال بالا را شبیه سازی کنید خواهید دید که با سویچ کردن کلید کنترلی این کلید در همان حالت می ماند

گاهی میخواهیم با زدن یک سویچ به اول حلقه برگردیم و سپس حلقه ادامه یابد و یا گاهی صرفا میخواهیم یک بافری را Refresh کنیم در این گونه مواقع به نظر شما ایده چیست ؟

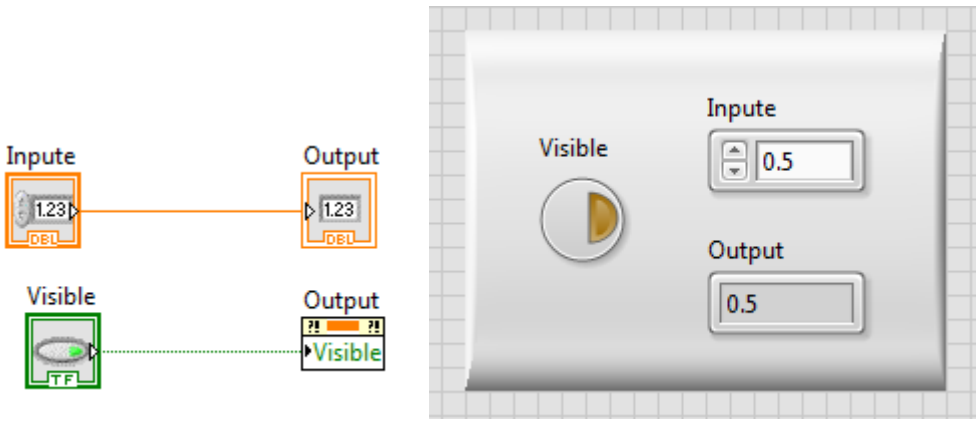
می توان یک برنامه برای این منظور نوشت و یک delay برای کلید تعریف کرد. اما لازم نیست چون...

روی کلید کنترلی کلیک راست کنید و از منوی باز شده Mechanical Action را انتخاب کنید. و گزینه ی مورد نظر خود را انتخاب کنید. و یا از منوی باز شده به قسمت properties رفته . به این ترتیب پنجره ای باز خواهد شد. و از پنجره ی باز شده Operation را انتخاب کنید :

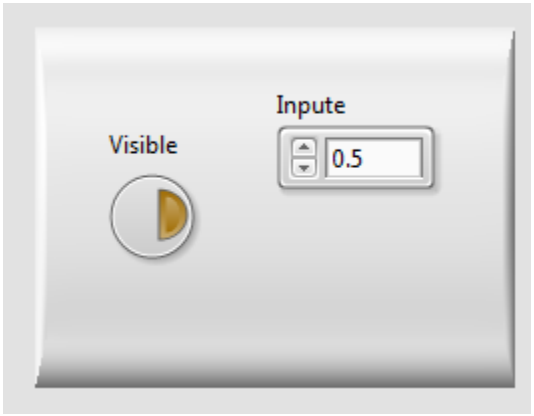
مثال : برنامه ای بنویسید که از ورودی سینوس بگیرد .



مثال : برنامه ی زیر را بررسی کنید .



توجه کنید که اگر کلید Visible زده نشود خروجی به صورت زیر خواهد بود :



به این ترتیب میتوانیم برخی از المان های صفحه ی نمایش را نامرئی کنیم و در مواقع ضرور آنها را مرئی کنیم.

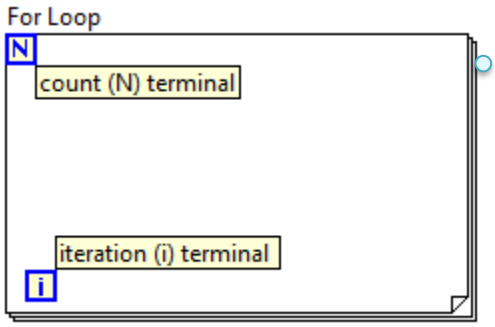
توجه : برای نوشتن این برنامه روی آیکن Output کلیک راست کنید. از منوی باز شده Create را انتخاب کنید.به این ترتیب منویی باز خواهد شد. از منوی باز شده Property Node را انتخاب کنید و از منوی باز شده Visible را انتخاب کنید.

روش دوم : میتوانید از پالت Application Control هم به Property Node دسترسی داشته باشید. اما اگر با این روش این تابع رابیاورد باید روی آن کلیک راست کنید و Output را Property Node ، link کنید.

اغلب دربرنامه ها لازم میشود که قسمتی از کد تکرار شود labview برای این منظور دو حلقه ی پیشنهاد میکند. پس میتوانید از for loop یا while loop استفاده کنید . حلقه ی for برای مواقعی است که تعداد تکرار حلقه مشخص است. اما حلقه ی while تاوقتی تکرار میشود که شرط حلقه (درست یا نا درست) بر قرار باشد.

For Loop

Executes its subdiagram n times, where *n* is the value wired to the count (**N**) terminal. The iteration (**i**) terminal provides the current loop iteration count, which ranges from 0 to *n*-1.



ازاین حلقه وقتی استفاده میکنیم که تعداد تکرار را میدانیم.

حلقه به تعداد دفعاتی اجرا میشود (یا تکرار) میشود که به N متصل شده باشد.

اجرای حلقه از صفر شروع میشود تا N-1

به برنامه ای که داخل حلقه قرار دارد subdiagram گفته میشود .

توجه : اگر به N صفر متصل کنید حلقه اجرا نمیشود.

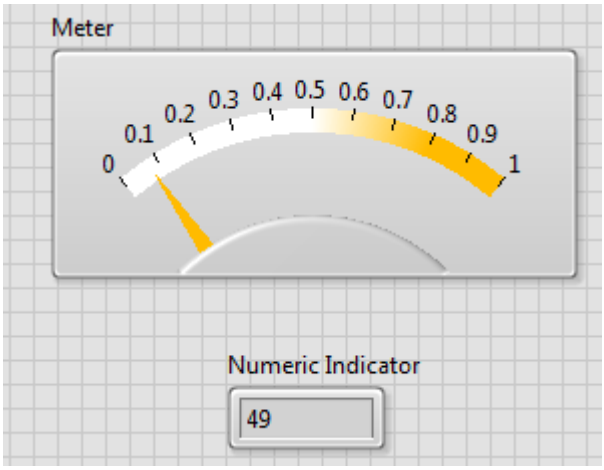
توجه : روند اجرای حلقه ی for به صورت زیر است:

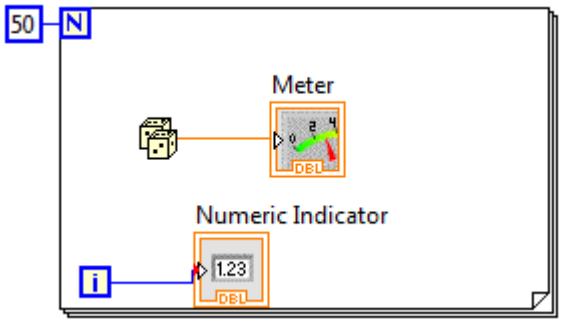
```
For (i=0 : N-1)

{
Execute subdiagram

}
```

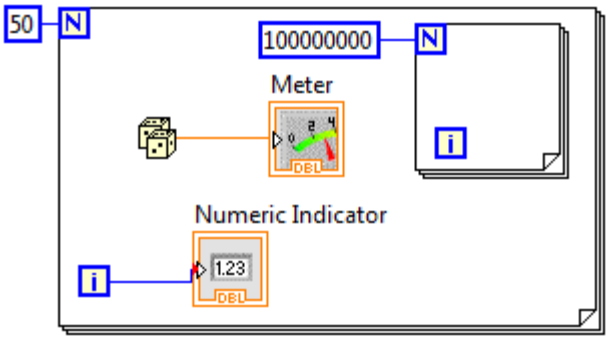
مثال : برنامهی زیررا بررسی کنید :





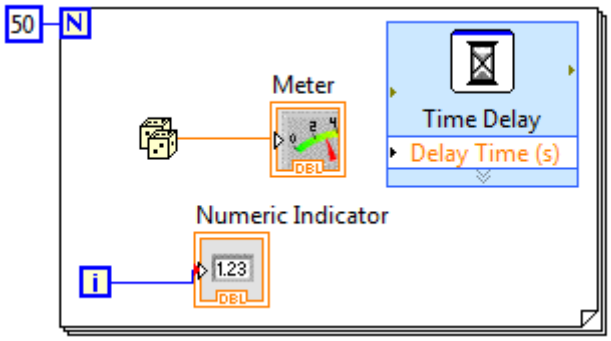
ملاحظه میکنید که اجرای حلقه خیلی سریع است و نمی توان چگونگی اجرای حلقه را دید. پس باید به یک روشی اجرای حلقه کند شود :

مثال :



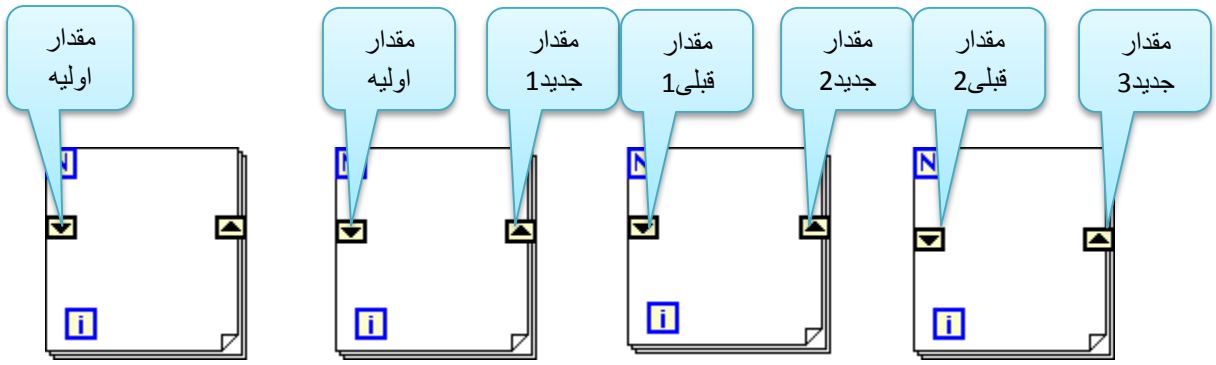
بازای هر اجرای حلقه ی بیرونی حلقه ی داخلی 100000000 بار اجرا میشود.

مثال : استفاده از تابع delay :

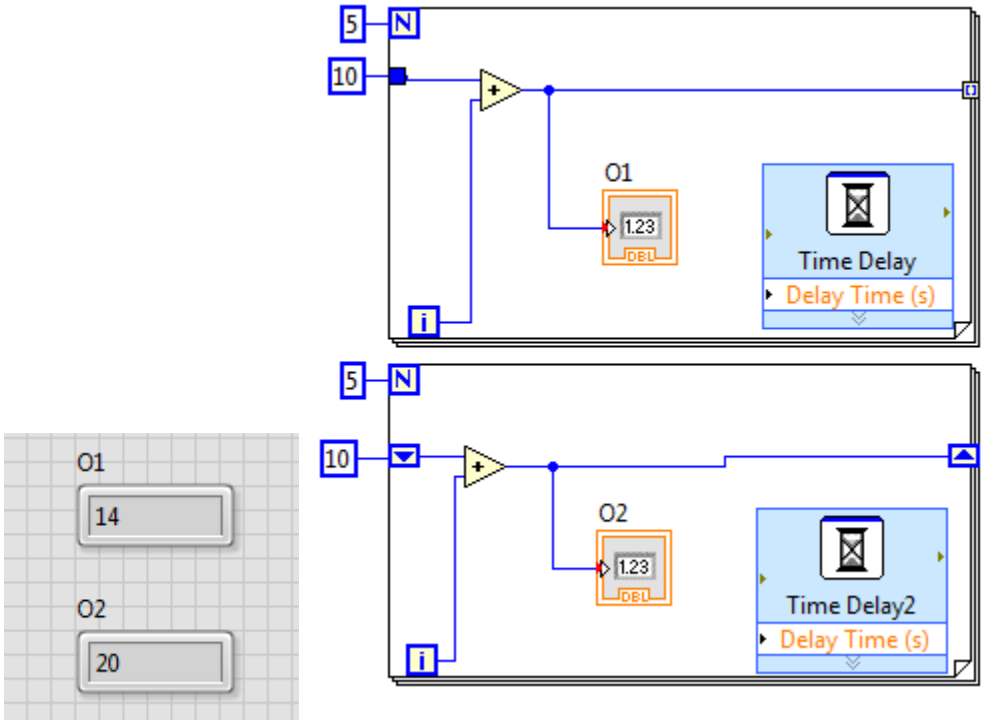


سوال : چگونه میتوان از یک تکرار حلقه به تکرار بعدی در حلقه ی for دیتایی را انتقال داد؟ فرض کنید قرار باشد از حلقه ی 5 به حلقه ی 6 دیتایی فرستاده شود . این کار با استفاده از Shift Register امکان پذیر است.

در واقع کار Shift Register این است ورودی راگرفته و صبر میکند که اجرای حلقه تمام شود و آن رابه خروجی میبرد. برای دست یابی به shift registerروی حلقه کلیک کنید و Add shift register را بزنید.



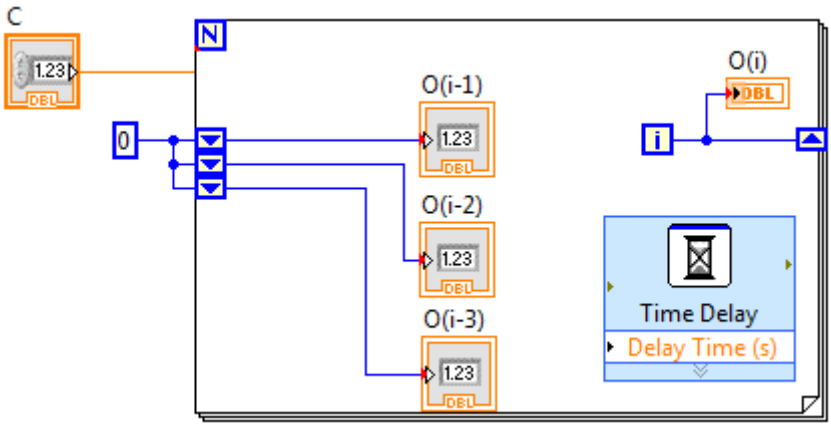
مثال : برنامه ی زیر عملکرد Shift Register را نشان میدهد.

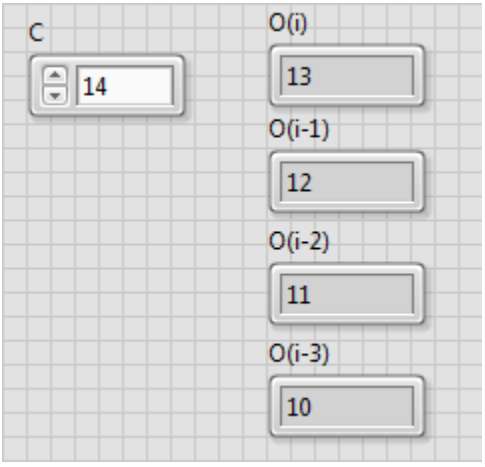


تحلیل :دراولین اجرای حلقه i=0 است . و در نتیجه خروجیadder نیز 10 است و این 10 به ترمینال سمت راست انتقال میابد. و در ابتدا ی حلقه ی بعد(یعنی وقتی که i=1 شده است). محتوای ترمینال سمت راست به ترمینال سمت چپ انتقال میابد. بعد از اتمام اجرای حلقه , خروجی adder برابر خواهد بود با 11 و این 11به ترمینال سمت راست منتقل میشود و در ابتدای اجرای حلقه ی سوم(یعنی وقتی که i=2 است.) محتوای ترمینال سمت راست به ترمینال سمت چپ انتقال میابد. و خروجی adder میشود 13و این 13 به ترمینال سمت راست منتقل میشود. و در ابتدای حلقه ی چهارم(یعنی وقتی i=3)به ترمینال سمت چپ منتقل خواهد شد. به این ترتیب خروجی adder 16خواهد شد و این 16 به ترمینال سمت راست منتقل خواهد شد. و در ابتدای حلقه ی پنجم (یعنی وقتی که i=4 شده است) محتوای ترمینال سمت راست که 16 است به ترمینل سمت چپ منتقل خواهد شد . به این ترتیب خروجی adder 20 خواهد بود .

نکته : اگر به مقادیر حلقه های قبل تر هم احتیاج شد روی ترمینال مورد نظر کلیک راست کنید و گزینه ی Add Element را انتخاب کنید.

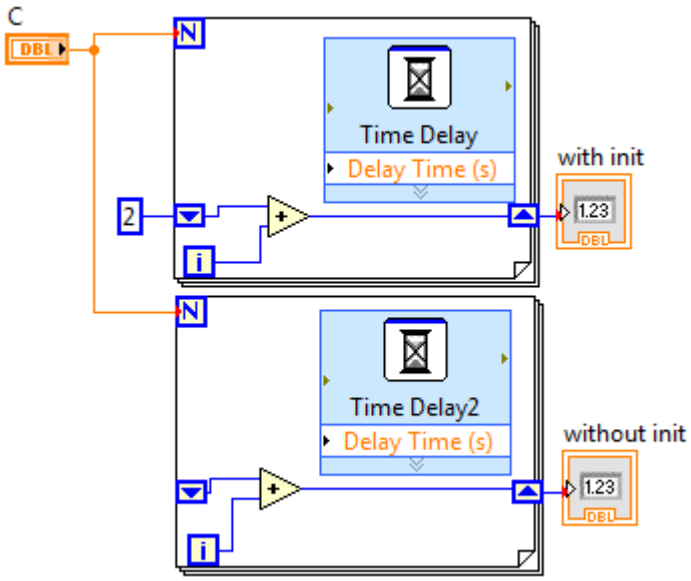
مثال : برنامه ی زیر را بررسی کنید :

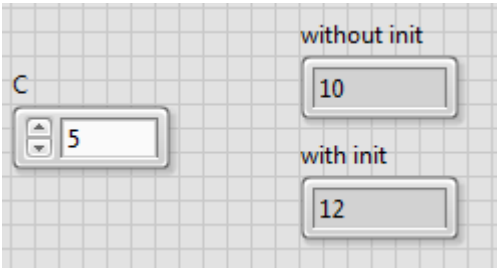




تذکر : بهتر است همیشه یک مقدار اولیه ای به Shift Register داده شود. تا از خروجی نامعلوم جلوگیری شود. اگر به Shift Register مقدار اولیه داده نشود. labview خودش یک عددی را به عنوان مقدار اولیه برای آن در نظر میگیرد.

مثال : برنامهی زیر را بررسی کنید :





توجه شود که خروجی With init در اجرای اولیه 12 است . و اگرتا ابد هم برنامه را اجرا شود 12 خواهد ماند. اما خروجی مربوط به Without init در اولین اجرا 10 است و در دومین اجرا 20 ودر سومین اجرا 30 و به همین ترتیب ادامه خواهد داشت. دلیل این مسئله این است که labview دیتای ذخیره شده درShift Register را تا وقتی که برنامه ی مربوطه بسته نشود حذف نمیکند. و مقدار نهایی اجرای در اجرای بعدی به عنوان مقدار اولیه خواهد ماند.

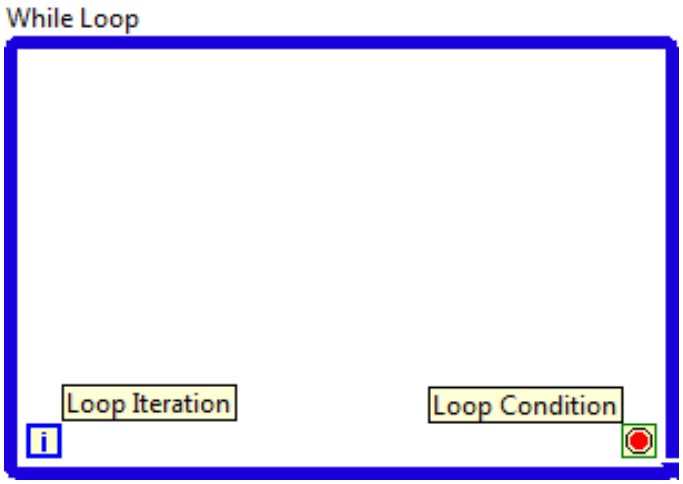
تذکر : گاهی لازم است به بازای درستی یک شرطی تکرار ادامه یابد و فقط وقتی حلقه متوقف شود که شرط نقض شده باشد یا شرط مورد نظر نادرست شده باشد.

تذکر : گاهی لازم است که بازای نادرستی شرطی تکرار ادامه یابد و وقتی که شرط درست شد حلقه متوقف شود.

While

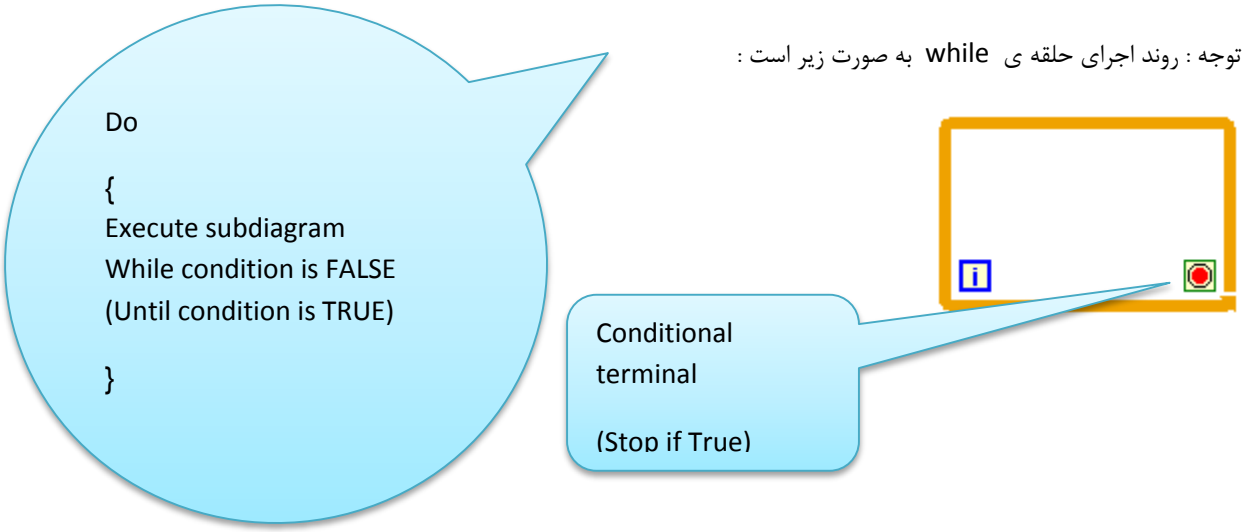
Repeats the subdiagram inside it until the conditional terminal, an input terminal, receives particular Boolean value. The Boolean value depends on the continuation behavior of the While Loop. Right-click the conditional terminal and select **Stop if True** or **Continue if True** from the shortcut menu. You also can wire an error cluster to the conditional terminal, right-click the terminal, and select **Stop on Error** or **Continue while Error** from the shortcut menu.

The While Loop always executes at least once.

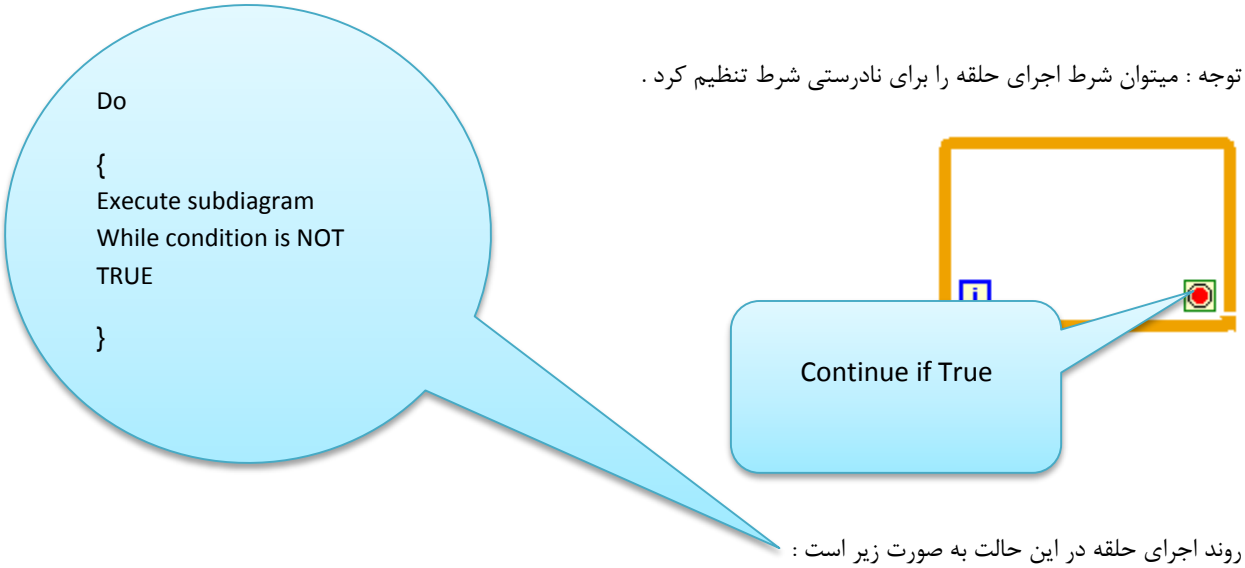


تعریف : حلقه ی while تا وقتی که عبارت جبری متصل به loop condition بر قرار باشد(درست باشد) اجرا میشود. و به محض اینکه شرط نقض شود برنامه از حلقه خارج خواهد شد.

توجه : روند اجرای حلقه ی while به صورت زیر است :

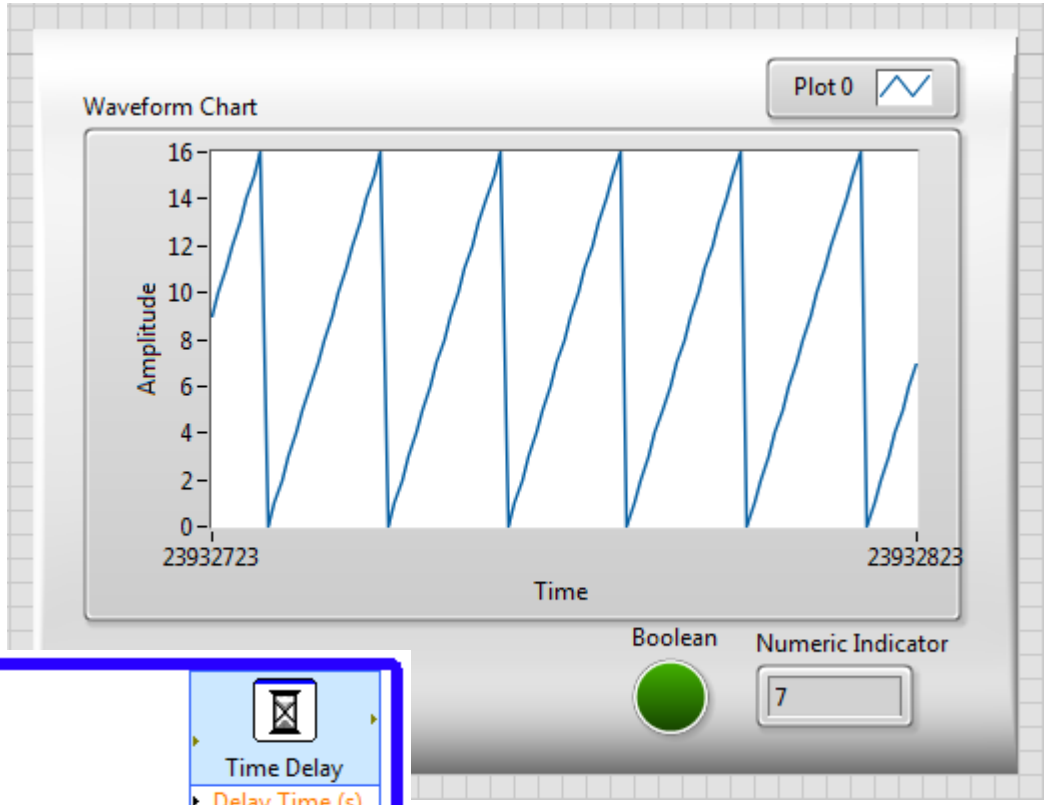


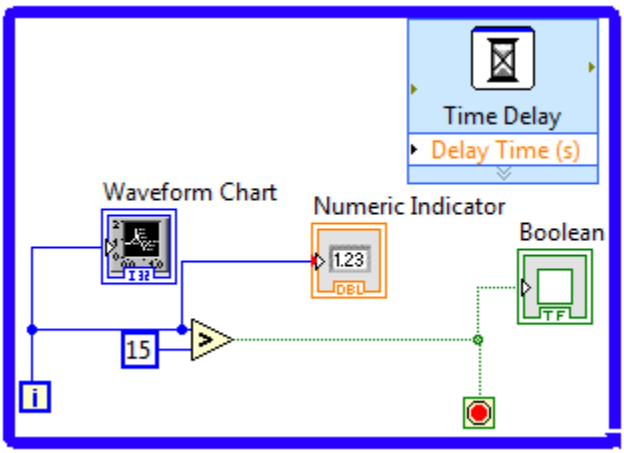
توجه : میتوان شرط اجرای حلقه را برای نادرستی شرط تنظیم کرد .



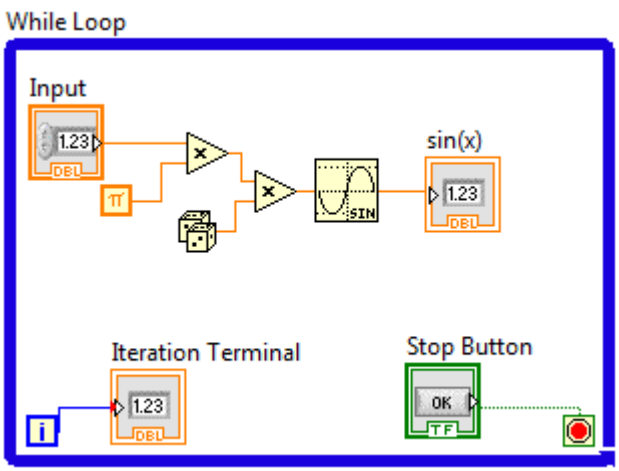
روند اجرای حلقه در این حالت به صورت زیر است :

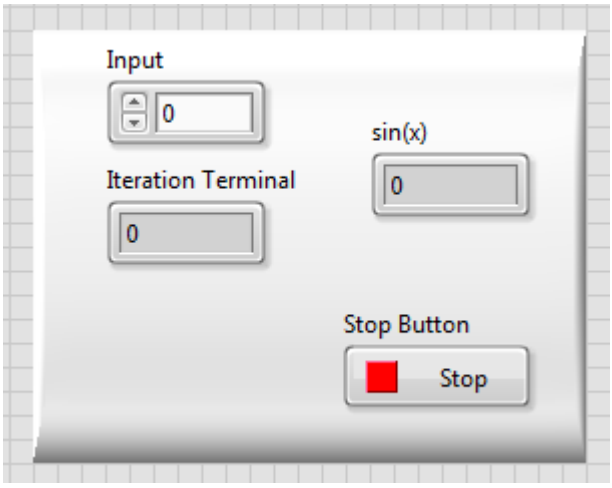
مثال : برنامه ی زیر را بررسی کنید :





مثال : برنامه ی زیر را بررسی کنید :



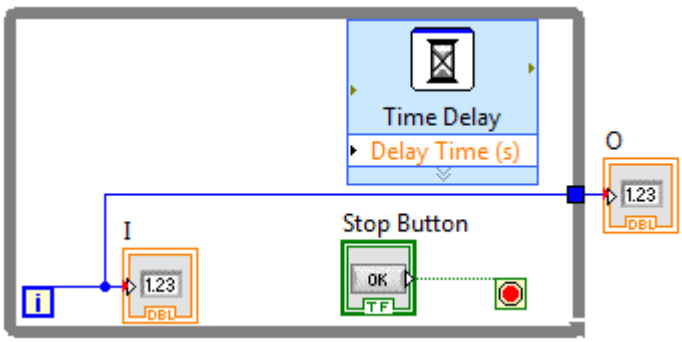


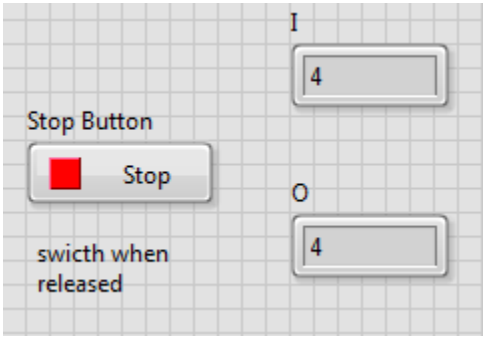
توجه : شروع اجرای حلقه از صفر است .

توجه : while در labview شبیه به do while در زبان C است پس حتی اگر شرط برقرار هم نباشد حداقل یک بار حلقه اجرا میشود. از این روی بایستی مواظب این مسئله باشید. برای مثال اگر برنامه ی فیدبک کنترل موتور dc را داخل یک حلقه ی while قرار داده باشیم و ولتاژ ورودی از حداکثر ولتاژ بیشتر شده باشد . باوجود این که شرط برقرار نیست ولی حلقه یک بار اجرا خواهد شد از این روی موتور و درایور های راه اندازی آسیب خواهند دید. البته توجه دارید که حلقه ی do while کلا به این صورت است که حلقه ی شرط را پس از اجرا بررسی میکند. که همان طور که گفته شد موجب مشکلاتی میشود که در مواقع حساس بایستی به نحوی شرط را چک بشود.

تذکر : آنچه را که در مورد Shift Register برای حلقه ی for گفته شد . در مورد while نیز درست است.

مثال : برنامه ی زیر را بررسی کنید :

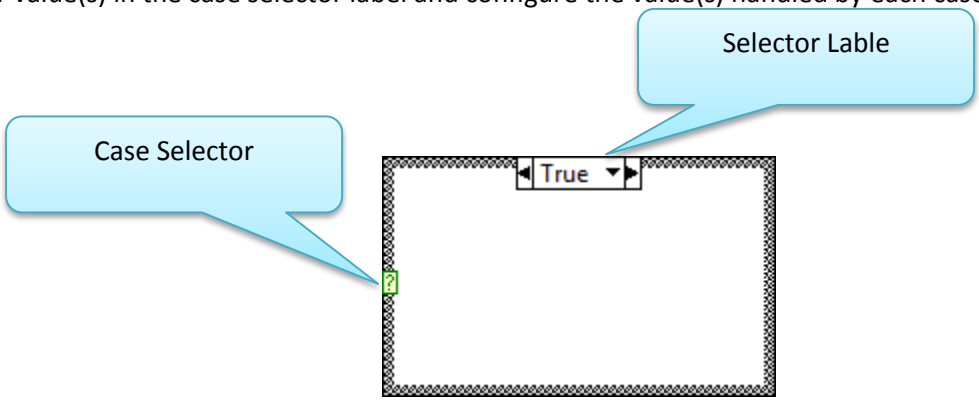




توضیح : نمایشگر O وقتی که حلقه متوقف میشود برابر با آخرین مقداری میشود که i دارد.

Case Structure :

Has one or more subdiagrams, or cases, exactly one of which executes when the structure executes. The value wired to the selector terminal determines which case to execute and can be Boolean, string, integer, enumerated type, or error cluster. Right-click the structure border to add or delete cases. Use the Labeling tool to enter value(s) in the case selector label and configure the value(s) handled by each case .



توجه: Case Structure دقیقا مثل if-else در زبان C هست .

توجه : ساختار شرطی میتواند چندین حالت باشد که هریک از شرط های وصل شده به Case Selector با Selector Lable برابر بود آن اجراخواهد شد.

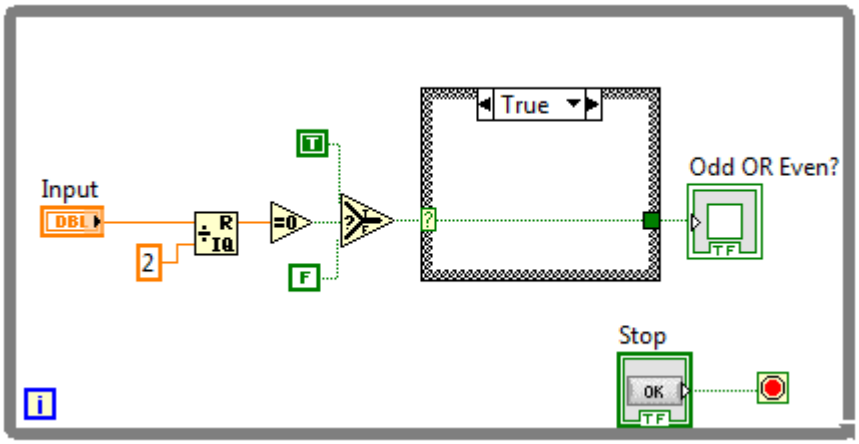
توجه : در صورت متصل کردن مقدار جبری به selector terminal حلقه ی Case دو حالت خواهد داشت True و Fulse .

توجه : چه مقادیری را میتوان به Case Selector متصل کرد ؟

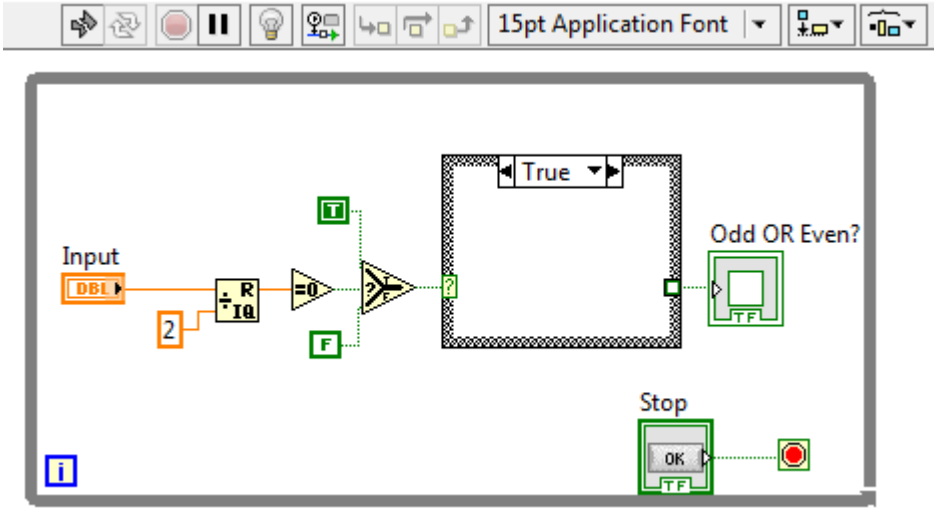
A Case Structure can accept one of five possible data types at its Case Selector terminal: Boolean, error cluster, integer, enum, and string. Each of the case structur below demonstrates how the case structure operates with each of these possible data types.

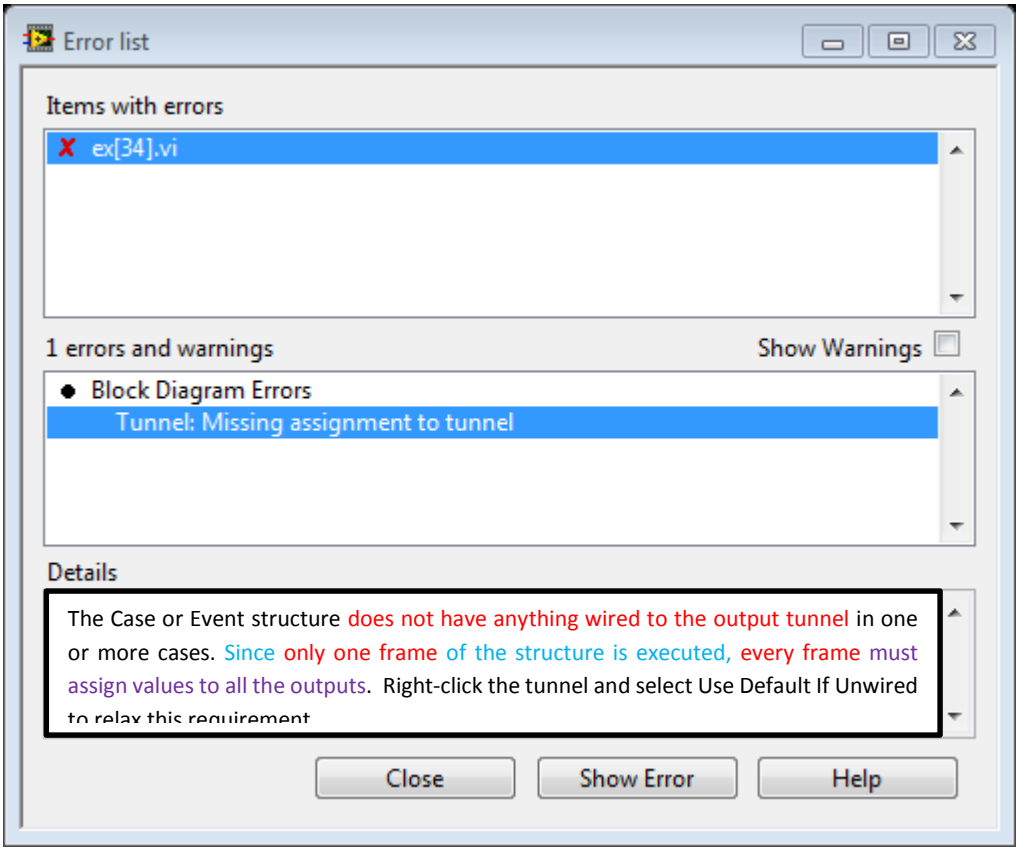
مثال : برنامه ای بنویسید که تشخیص دهد عددی که وارد میکنید زوج هست یا فرد ؟

راهنمایی : باقیمانده ی تقسیم یک عدد فرد به 2یک است و باقیمانده ی تقسیم یک عدد زوج به 2 صفر است.

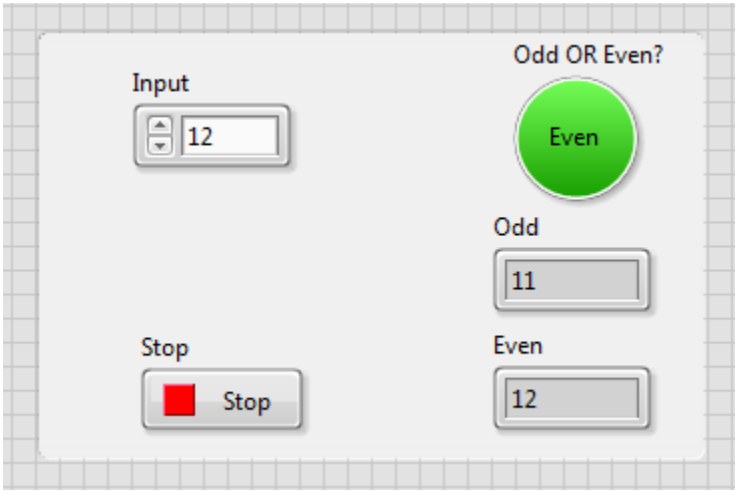


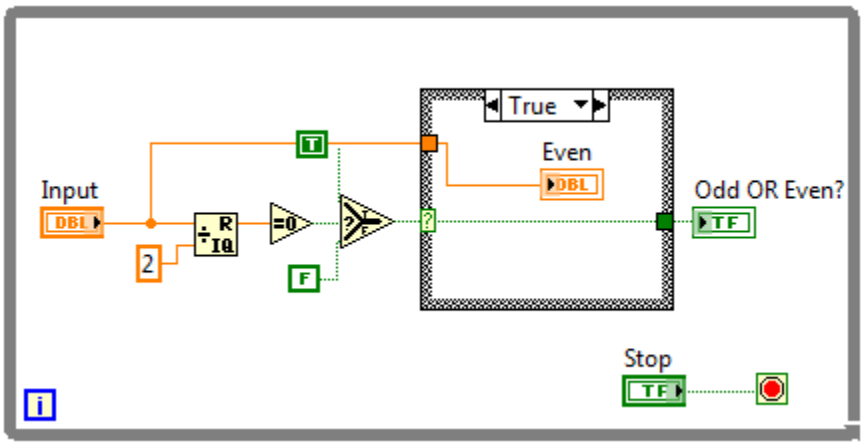
توجه : چنانچه در یک برنامه ای Case 10 داشته باشیم و اگر تنها یکی از Case ها به خارج از Case متصل شده باشد . باید از سایر Case ها هم به آن متصل شده باشد و الا error رخ میدهد. برای مثال در Case مربوط به حالت True اتصال به بیرون قطع بشود خطا رخ خواهد داد :

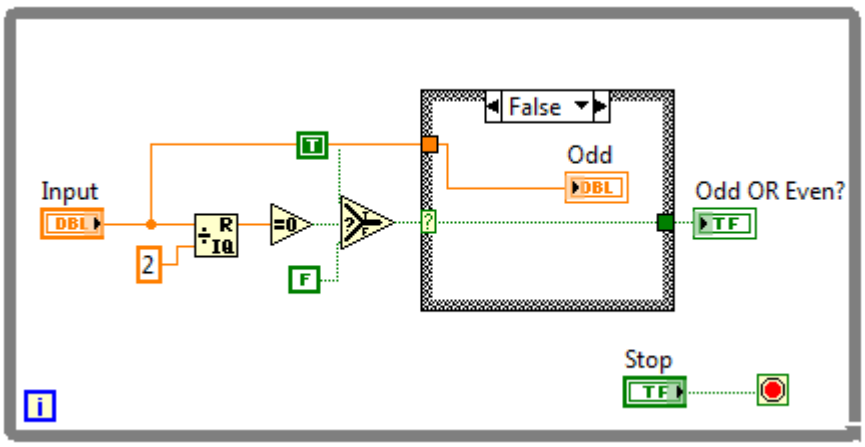




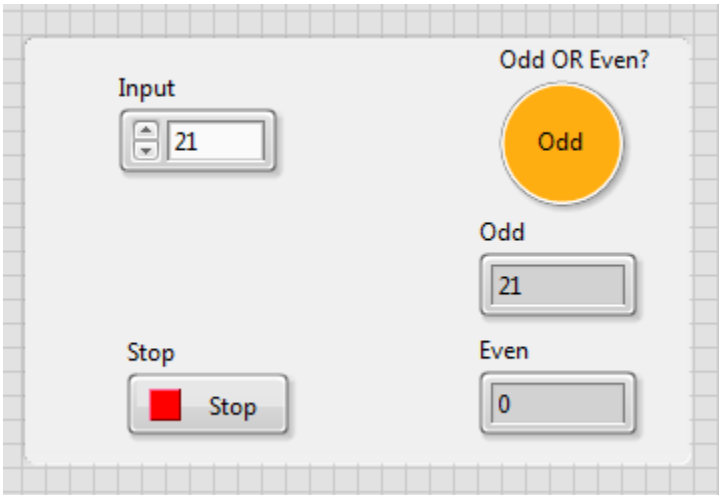
مثال : مثال زیر را بررسی کنید :

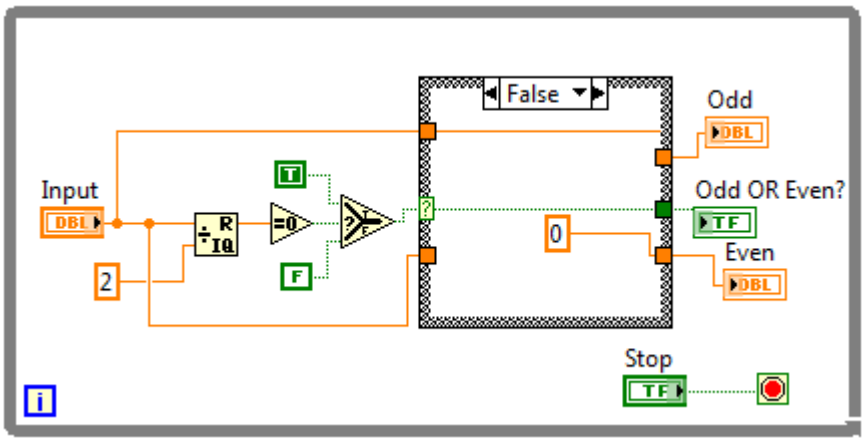


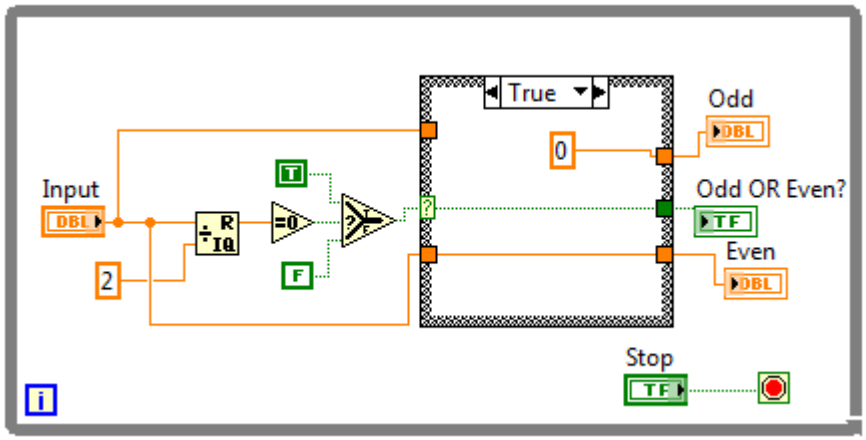




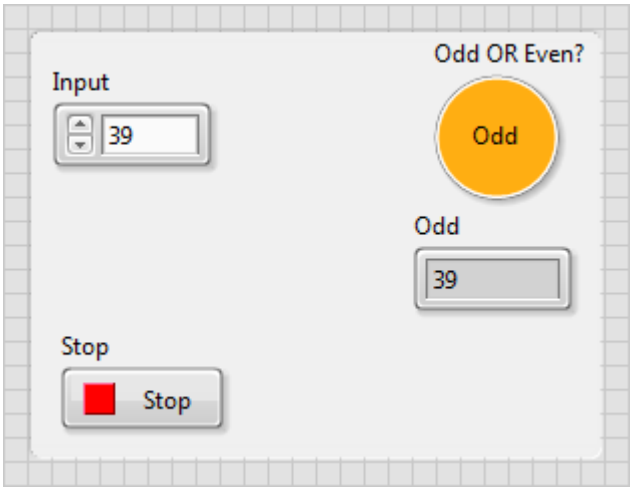
مثال : تفاوت مثال بالا با مثال زیر را کشف کنید :

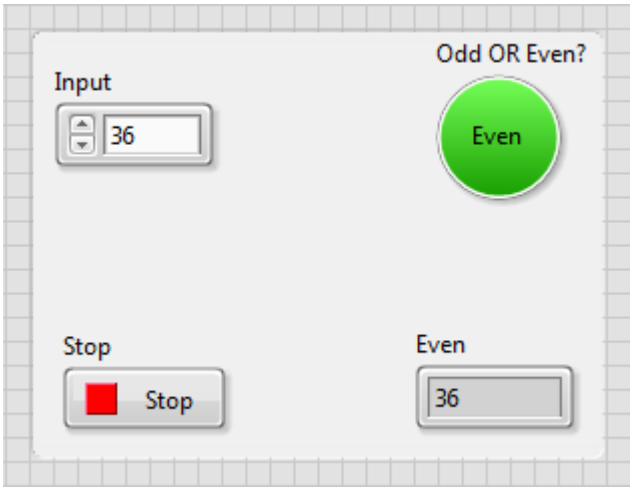


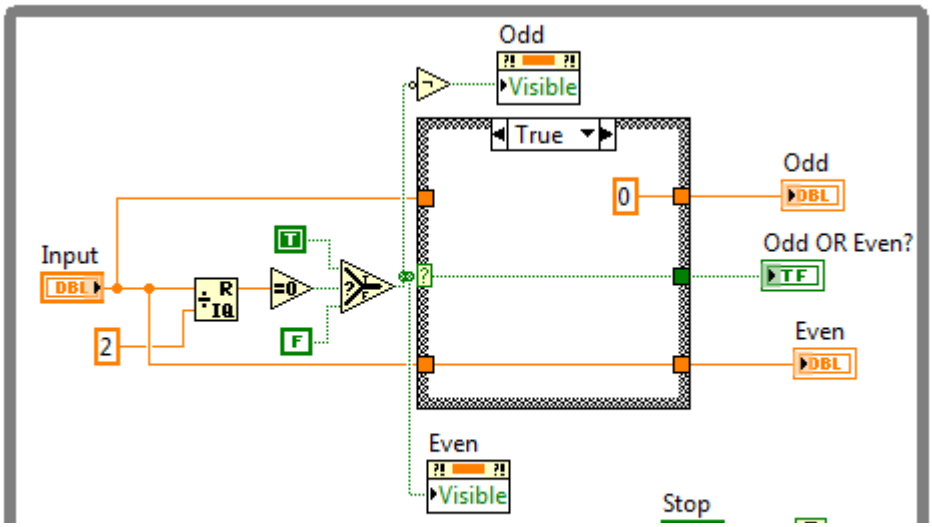


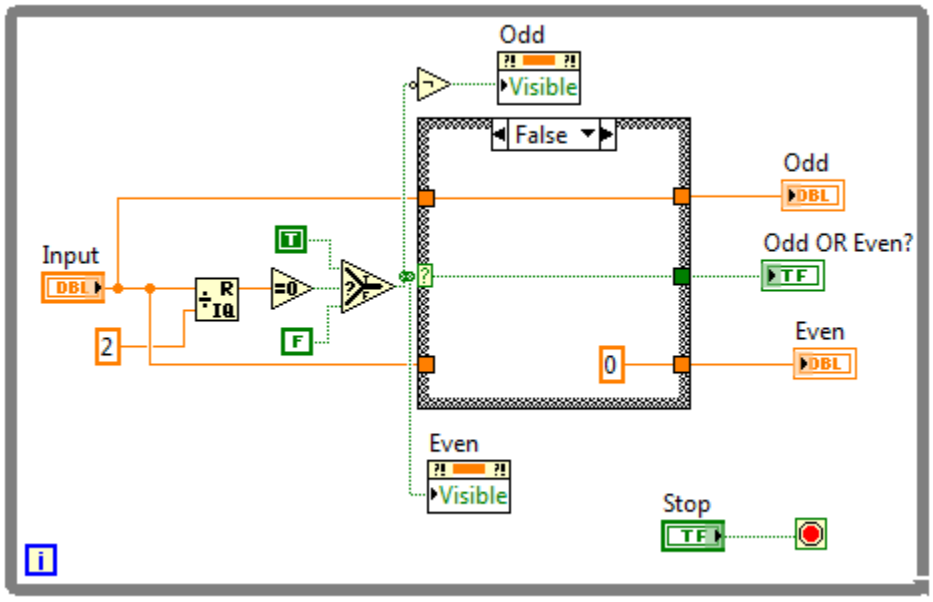


مثال : تفاوت مثال بالا با مثال زیر را کشف کنید :

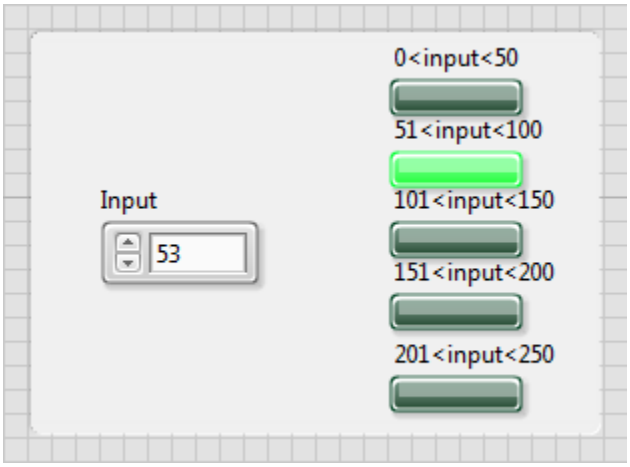

Input
39
Odd OR Even?
Odd
Odd
39
Stop
Stop

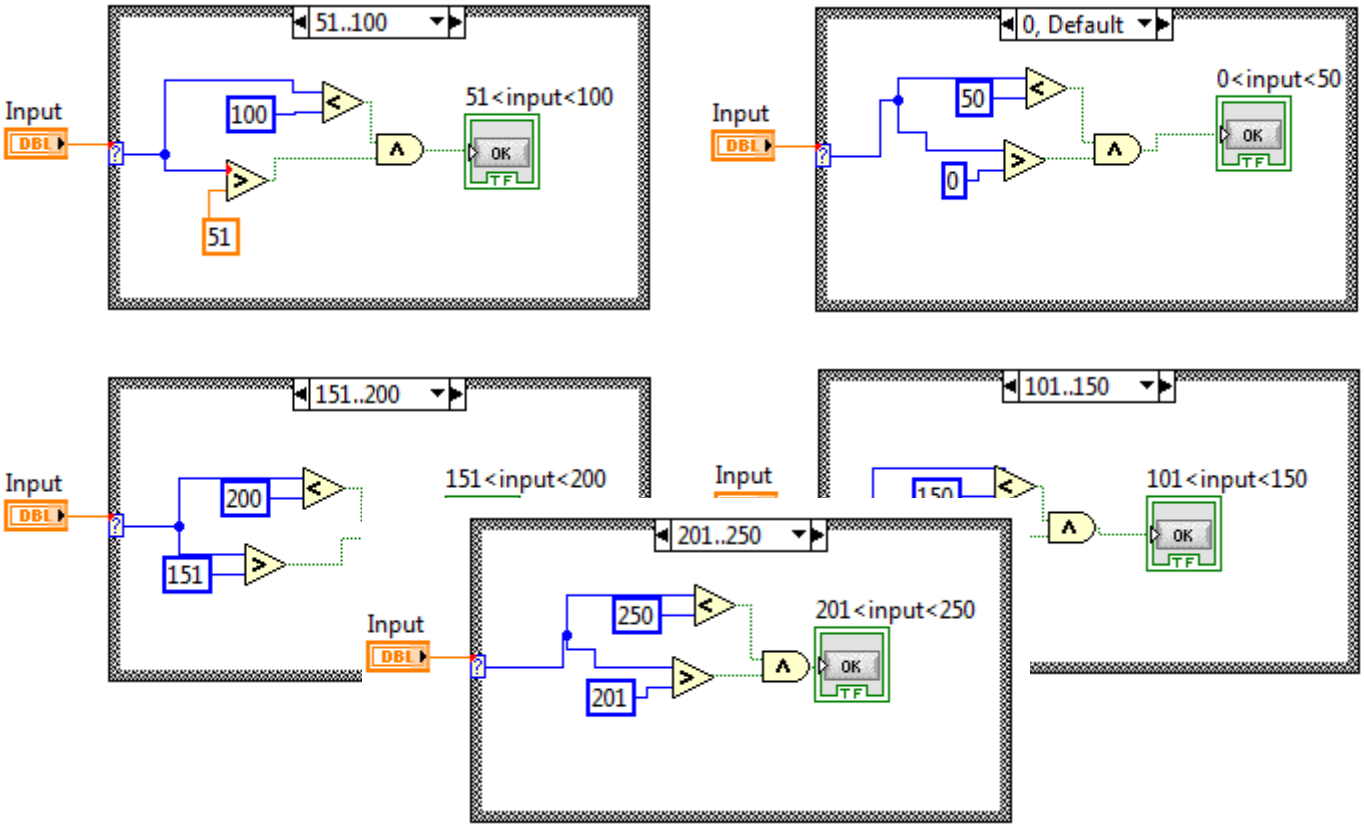

Input
36
Odd OR Even?
Even
Stop
Stop
Even
36


Odd
Visible
True
Odd
0
Input
DBL
T
=0
F
2
Odd OR Even?
TF
Even
DBL
Even
Visible
Stop

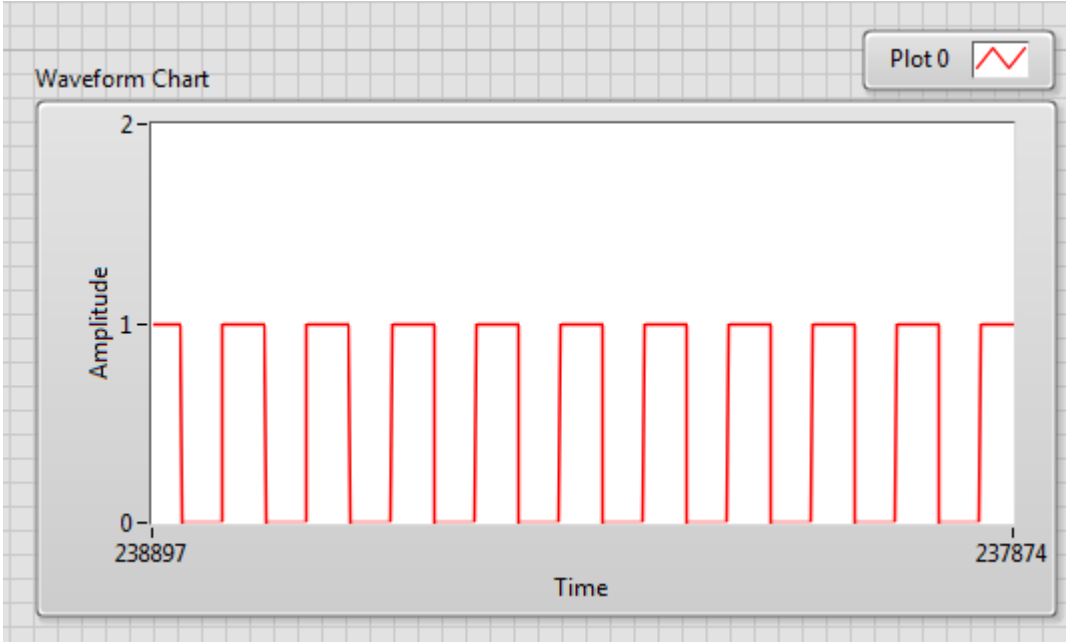

Odd
Visible
False
Odd
DBL
Input
DBL
T
=0
F
2
Odd OR Even?
TF
Even
0
Even
Visible
Stop
TF
i

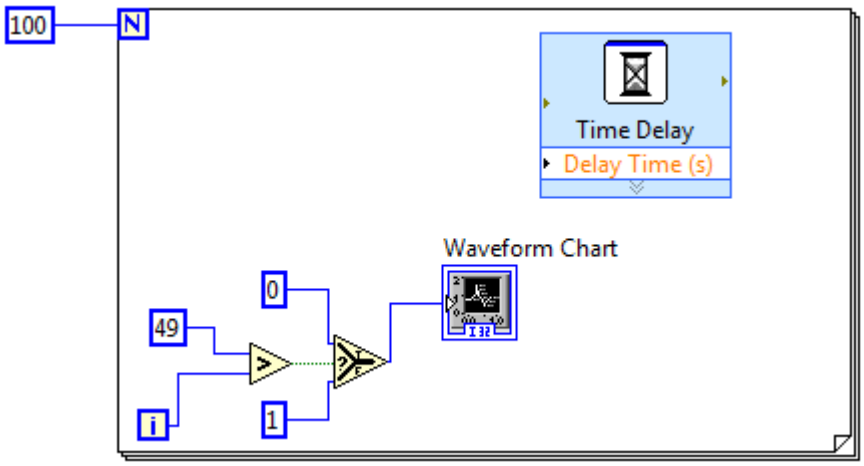
مثال : برنامه ی زیر را بررسی کنید :



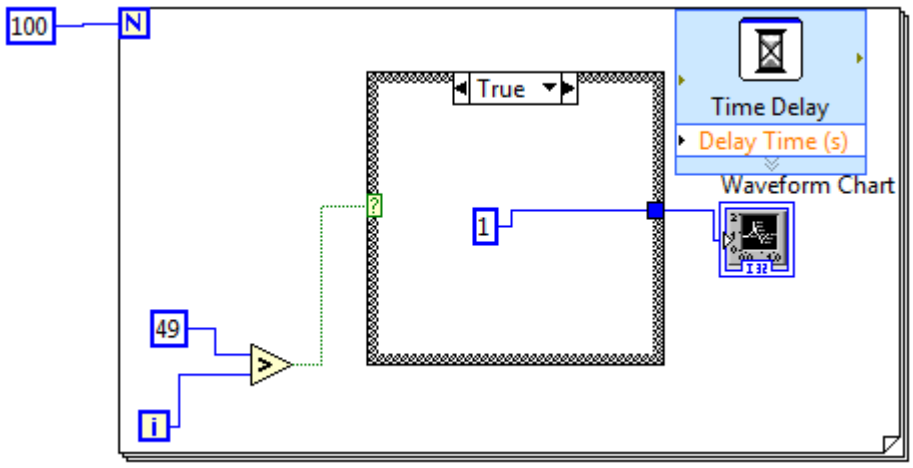


مثال : برنامه ای بنویسید که یک موج pwm با 50٪ duty cycle ایجاد کند :

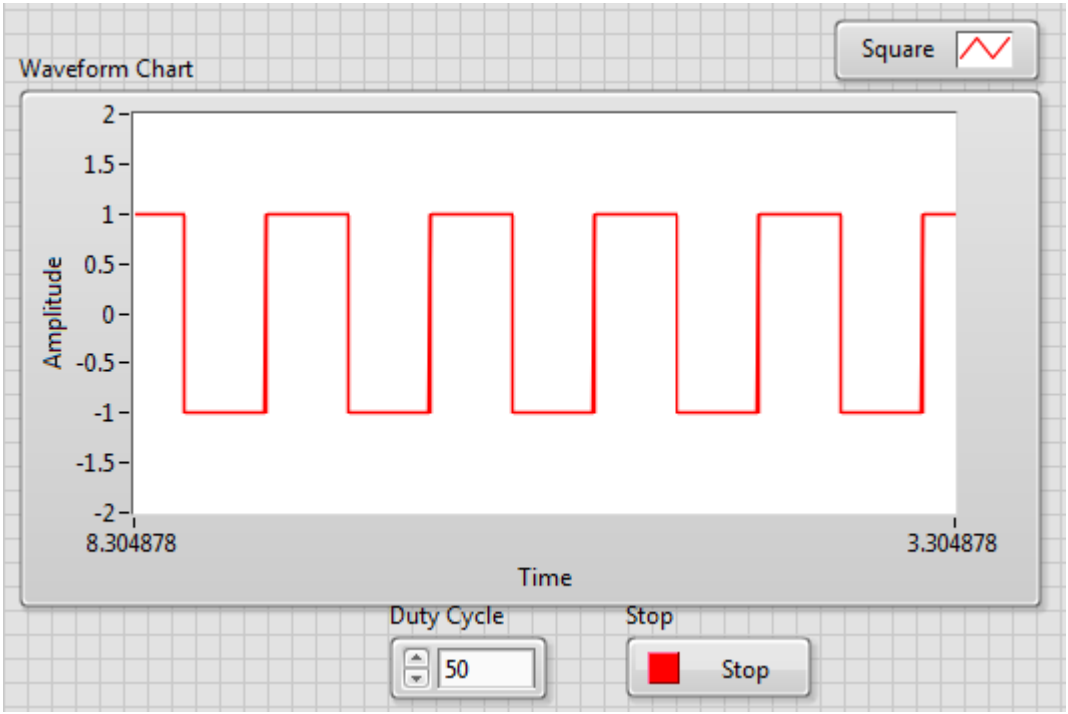


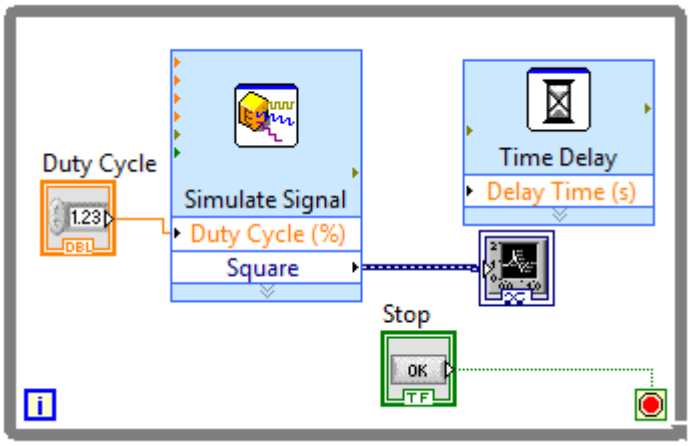


مثال : برنامه ی بالا را با Case بنویسید :

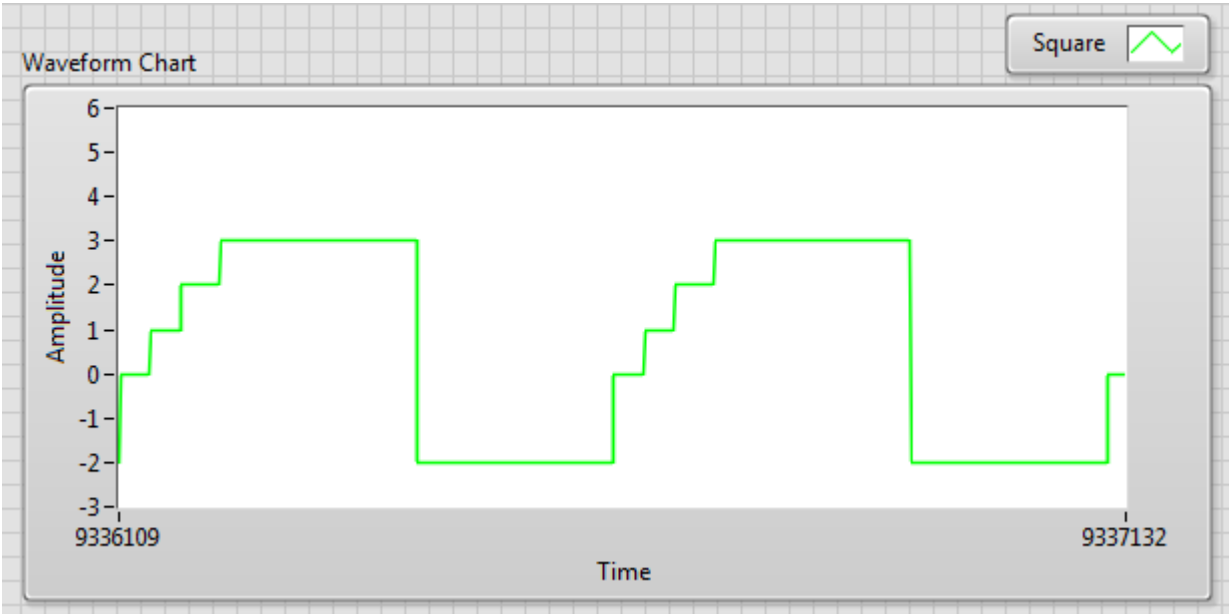


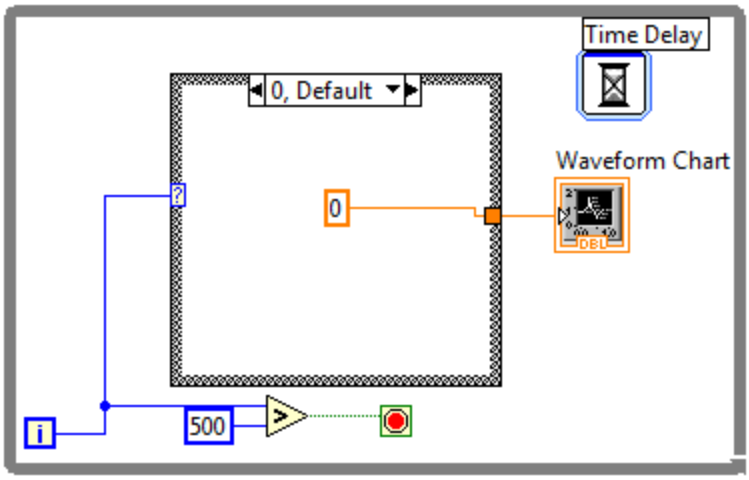
مثال : برنامه ی بالا را میشود به صورت زیر هم نوشت :

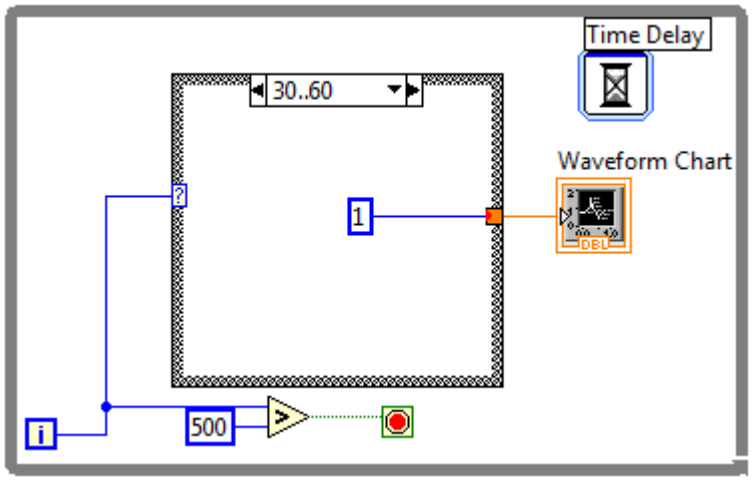


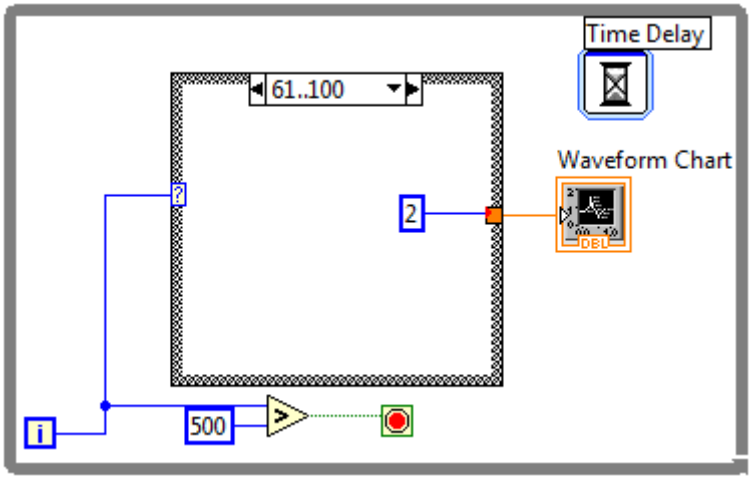


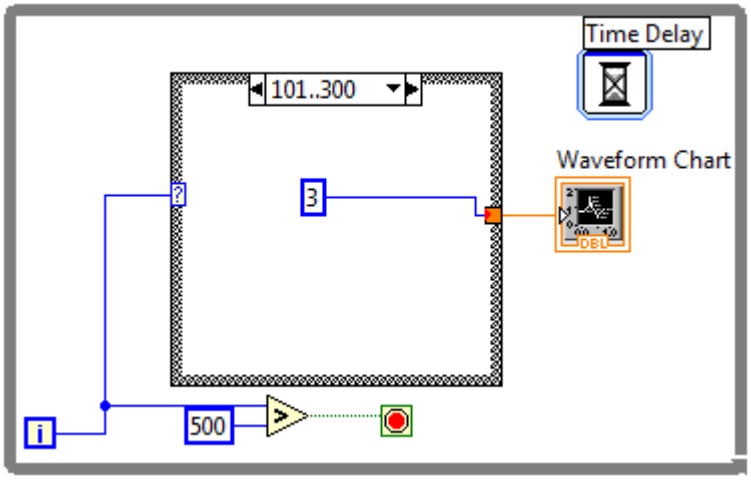
مثال : برنامه ی زیر را بررسی کنید :

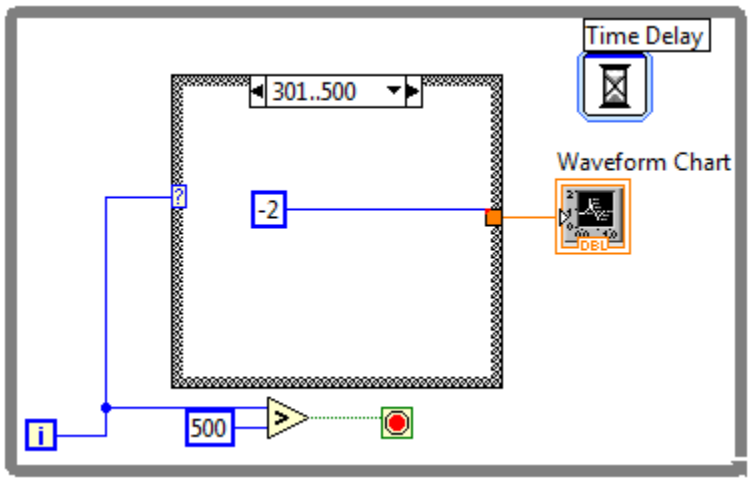




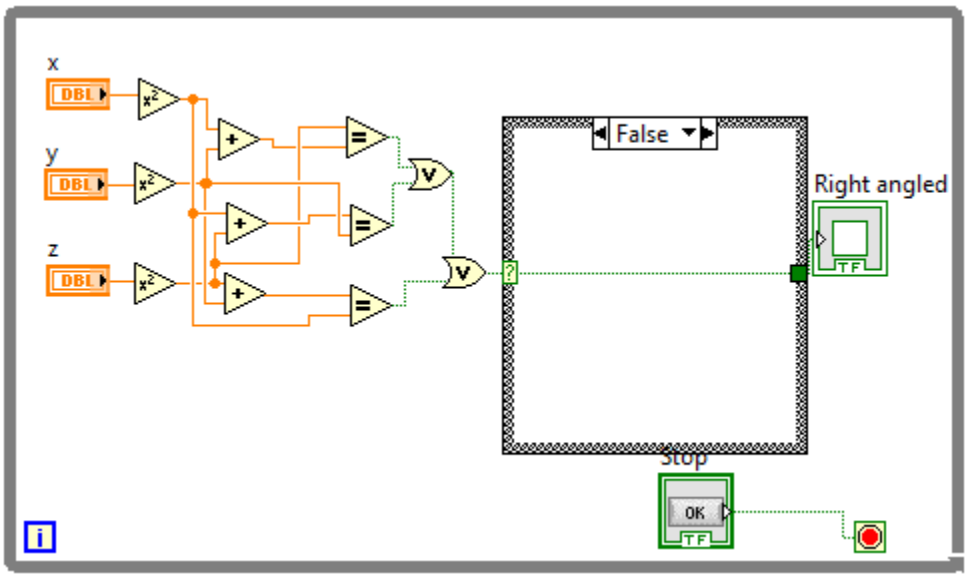

Time Delay
30..60
Waveform Chart
1
i
500


Time Delay
61..100
Waveform Chart
2
i
500

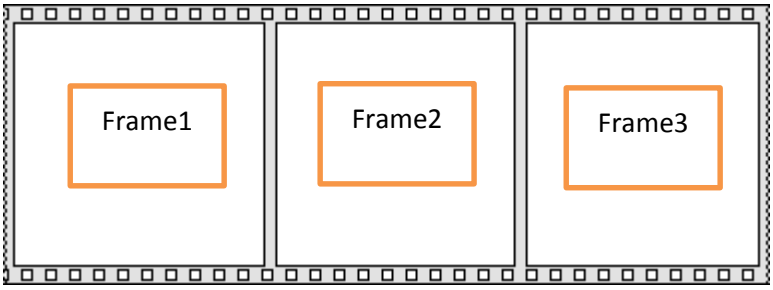

Time Delay
101..300
Waveform Chart
3
i
500



مثال : برنامه ای بنویسید که از شما سه ضلع بگیرد که و اگر مثلث قائم بود چراغی روشن شود.(البته این برنامه ایراد هایی هم دارد.)



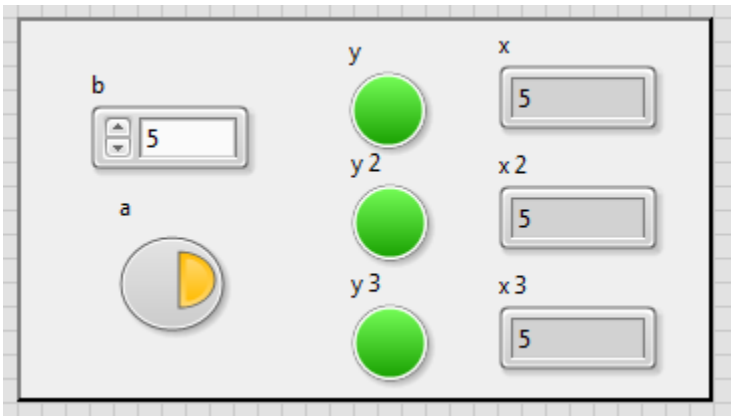
Flat Sequence Structure : ساختار ترتیبی

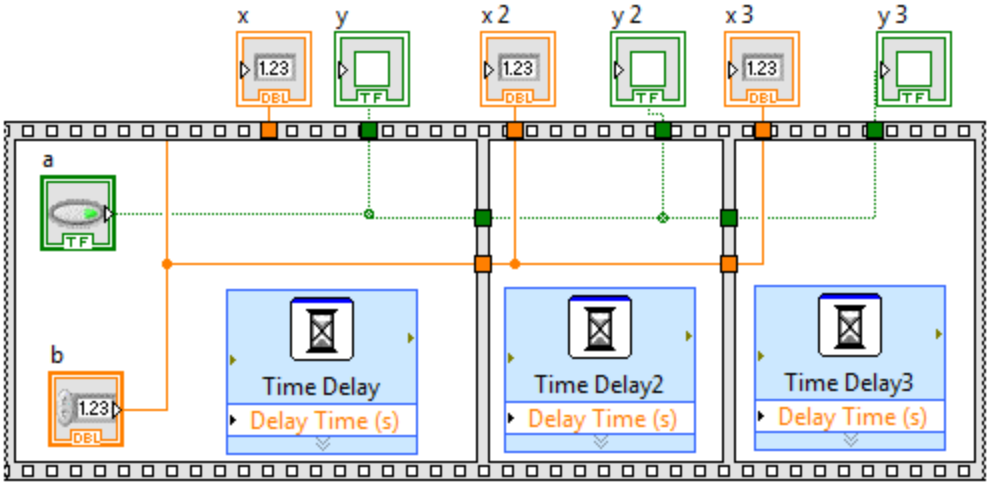


اگر قرار باشد اول بخشی از برنامه اجرا شود بعد و بعد بخش دیگری از برنامه از این ساختار استفاده میشود. در واقع با این ساختار میتوان ترتیب اجرا یا ترتیب توالی را تنظیم کرد.

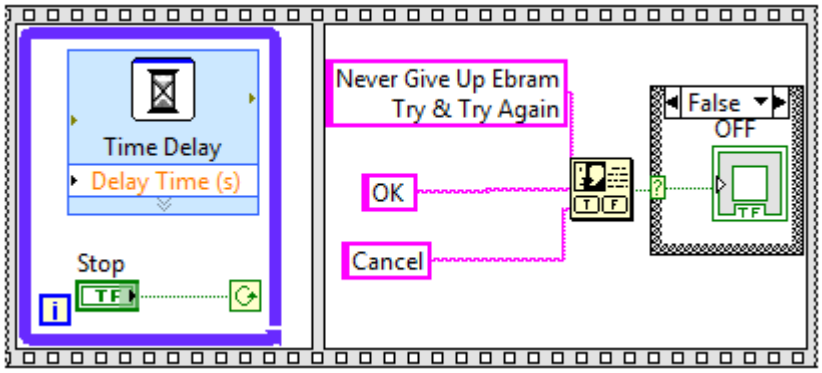
پس ابتدا Frame1 اجرا میشود بعد اتمام اجرای Frame1 ، Frame2 اجرا میشود. و در نهایت هم Frame3 اجرا خواهد شد.

مثال : برنامه ی زیر را بررسی کنید :

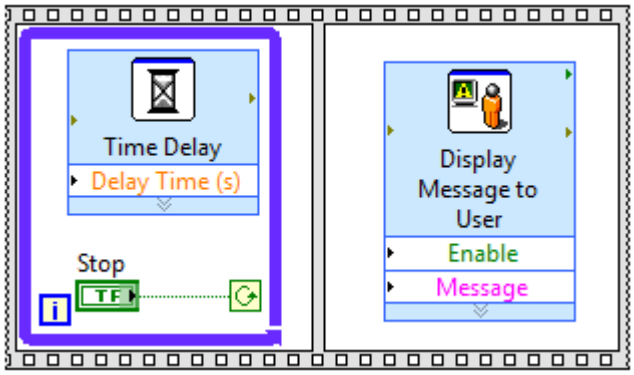




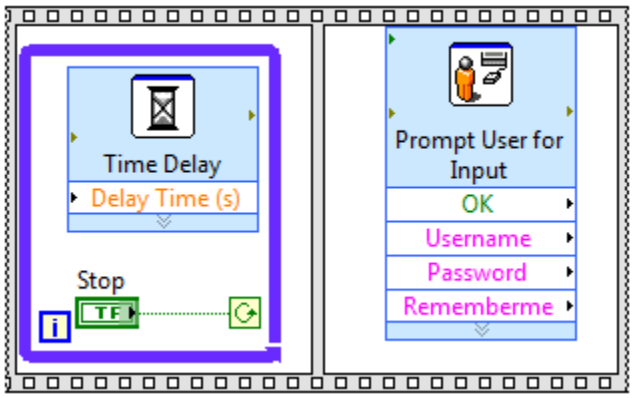
مثال : برنامه ی زیر را بررسی کنید : (توجه کنید که این برنامه را به صورت پیوسته اجرا نکنید)



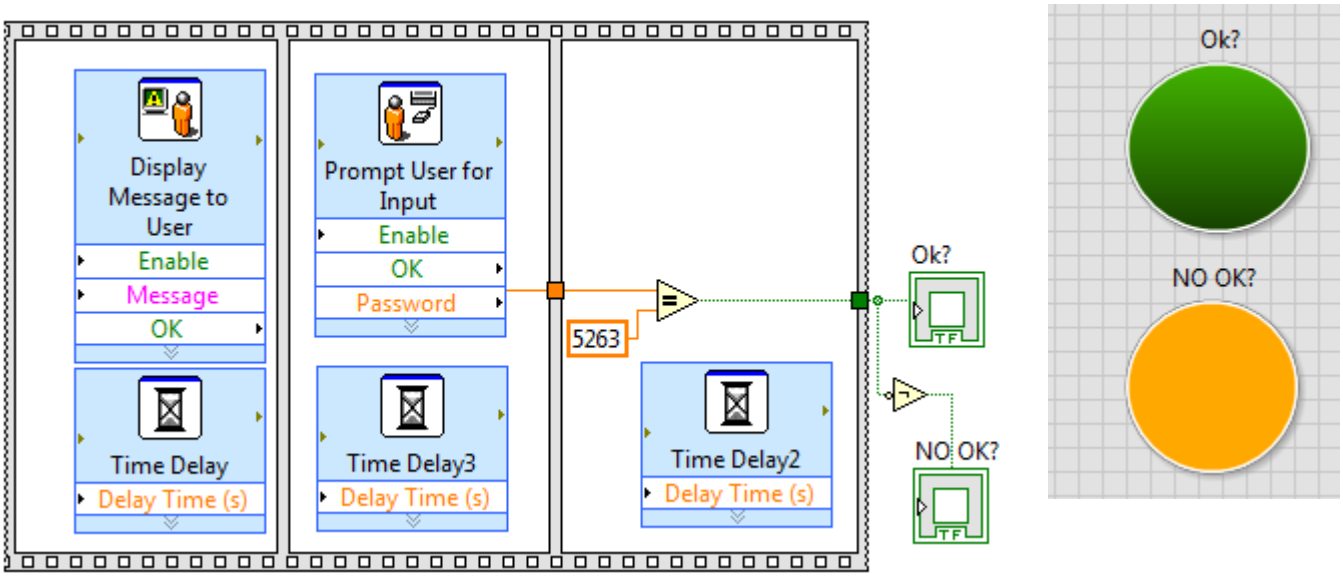
مثال : برنامه ی بالا را میشود به روش زیر هم نوشت :



مثال : برنامه ی بالا را میشود به روش زیر هم نوشت :



مثال : برنامه ای بنویسید که اگر پسورد ورودی درست بود یک LED روشن شود و اگر پسورد وارد شده نادرست بود LED خاموش شود



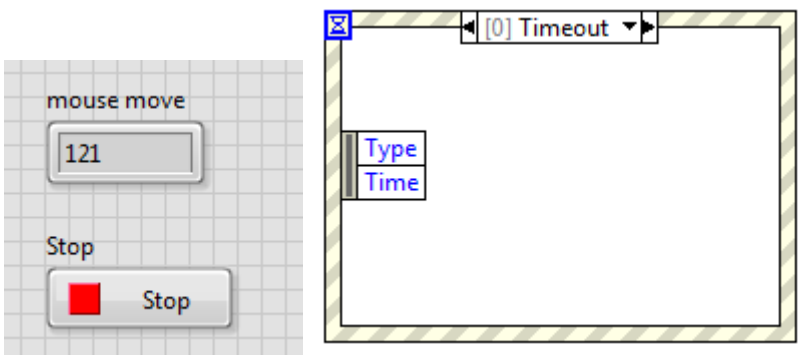
تذکر : پالتِ Dialog بعدا به طور کامل بررسی خواهد شد .

معرفی Event Structure :

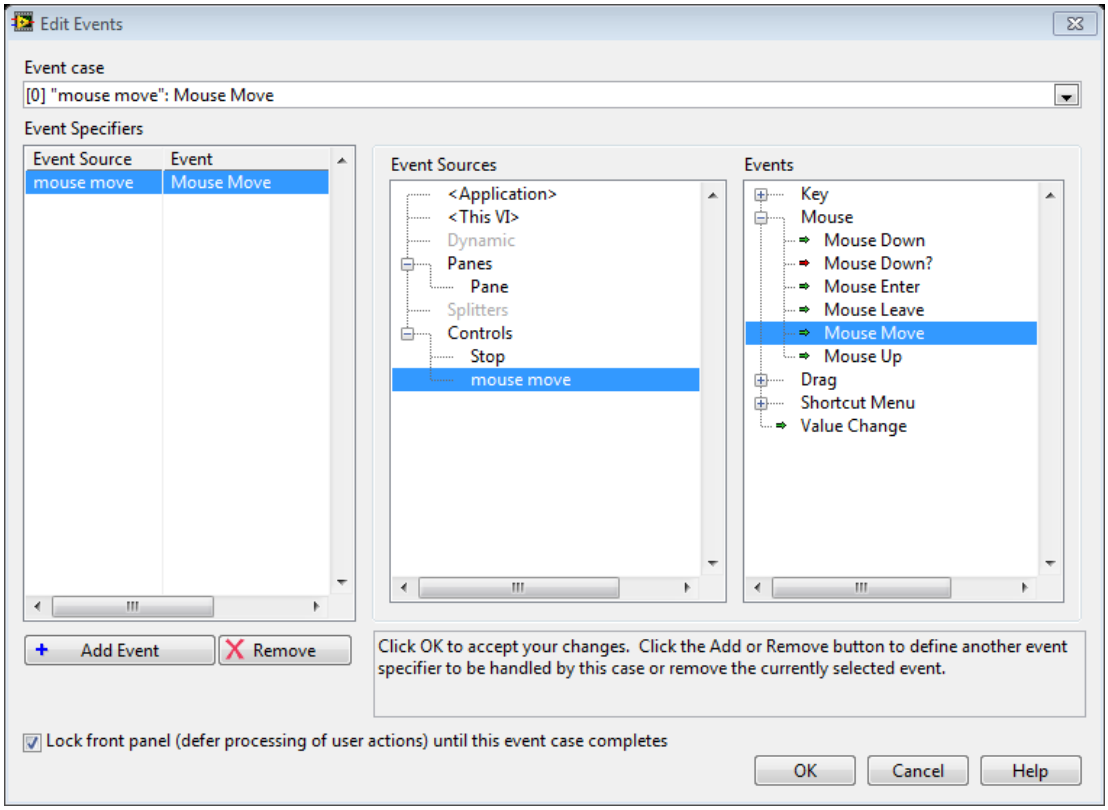
این ساختار را میتوان با وقفه مشابه دانست. در واقع مفهوم Event این است که در برنامه یک پرچم قرار داده میشود و در صورتی که اتفاقی که پیش بینی کرده شده بود اتفاق افتاد به اجرای ادامه ی برنامه پرداخته میشود.

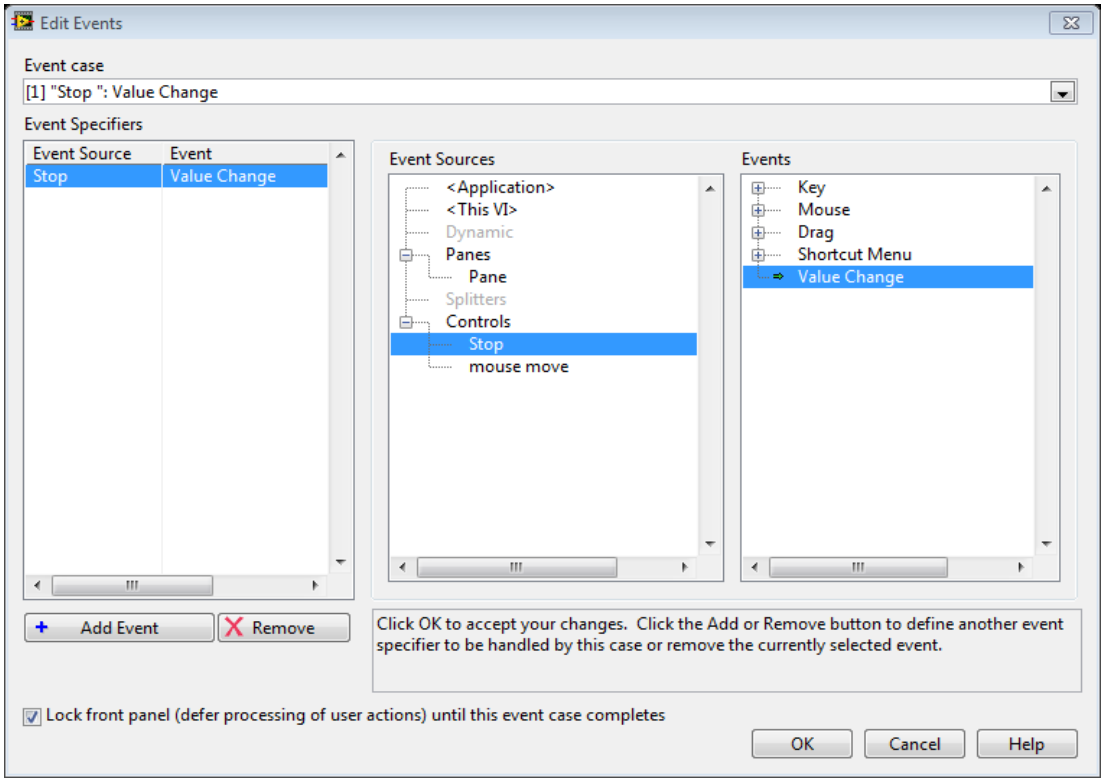
مثال : برنامه ی زیر را بررسی کنید :

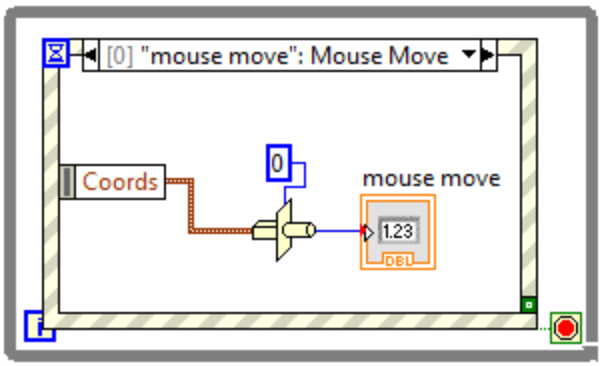
یک حلقه ی Event آورده شود :

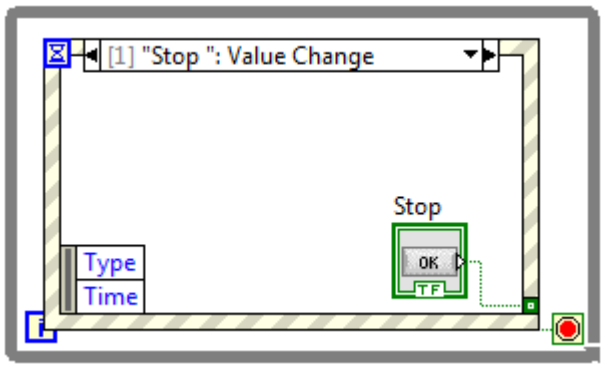


روی آن کلیک راست و از منوی باز شده Edit Event را انتخاب و مراحل بعدی به صورت زیر اعمال شود:


Edit Events
Event case
[0] "mouse move": Mouse Move
Event Specifiers
Event Source
Event
mouse move
Mouse Move
Event Sources
<Application>
<This VI>
Dynamic
Panes
Pane
Splitters
Controls
Stop
mouse move
Events
Key
Mouse
Mouse Down
Mouse Down?
Mouse Enter
Mouse Leave
Mouse Move
Mouse Up
Drag
Shortcut Menu
Value Change
Add Event
Remove
Click OK to accept your changes. Click the Add or Remove button to define another event specifier to be handled by this case or remove the currently selected event.
Lock front panel (defer processing of user actions) until this event case completes
OK
Cancel
Help

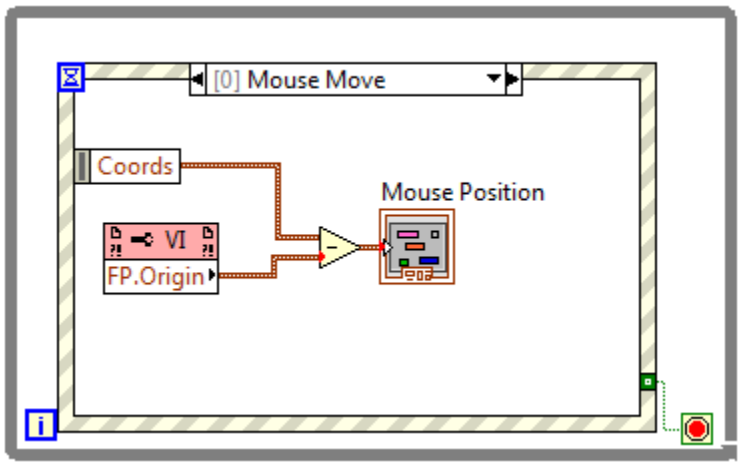

Edit Events
Event case
[1] "Stop ": Value Change
Event Specifiers
Event Source
Event
Stop
Value Change
Event Sources
<Application>
<This VI>
Dynamic
Panes
Pane
Splitters
Controls
Stop
mouse move
Events
Key
Mouse
Drag
Shortcut Menu
Value Change
Add Event
Remove
Click OK to accept your changes. Click the Add or Remove button to define another event specifier to be handled by this case or remove the currently selected event.
Lock front panel (defer processing of user actions) until this event case completes
OK
Cancel
Help

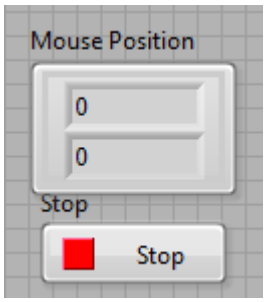


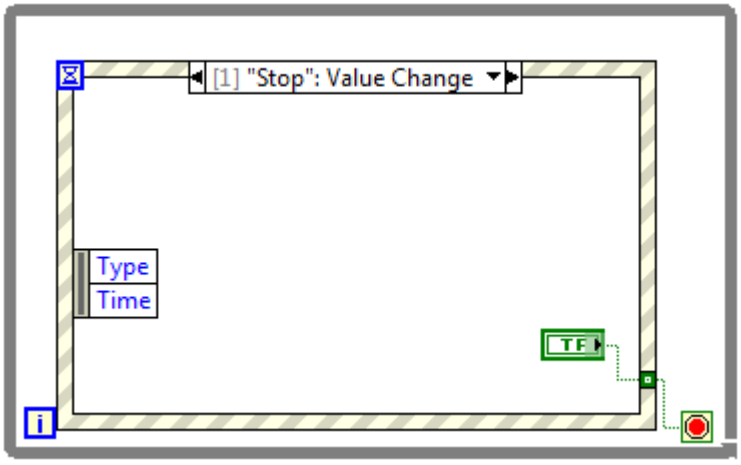


حالا اگر برنامه را اجرا و موس را تغییر دهید متوجه اجرای برنامه خواهید شد.

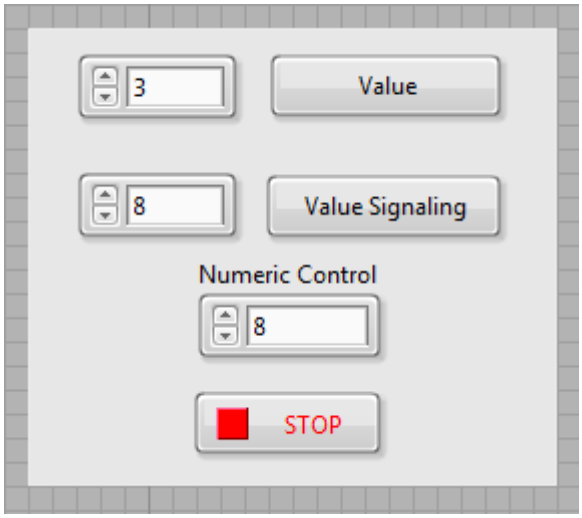
مثال : اگر برنامه را مثل زیر تغییر دهید بهتر اجرا خواهد شد :

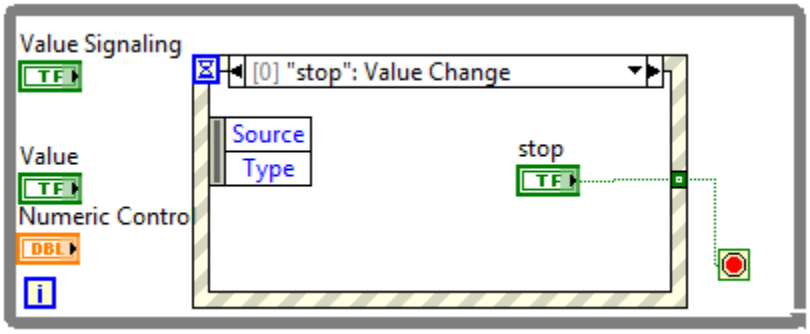


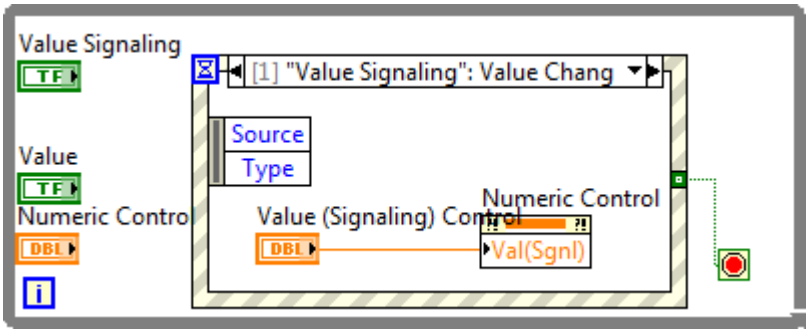


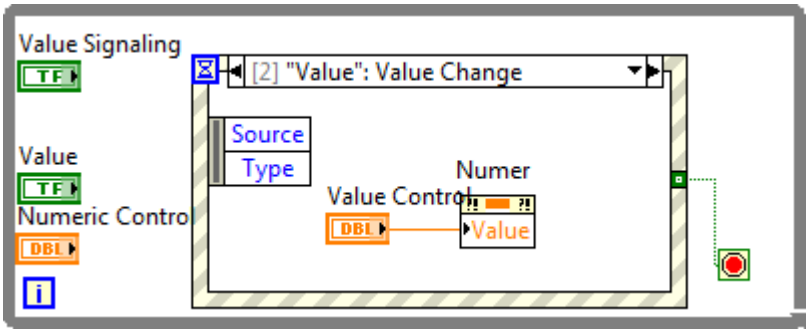


مثال : برنامه ی زیر را بررسی کنید :

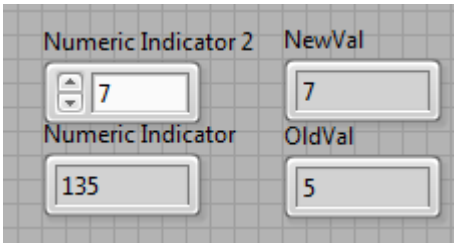

3
Value
8
Value Signaling
Numeric Control
8
STOP

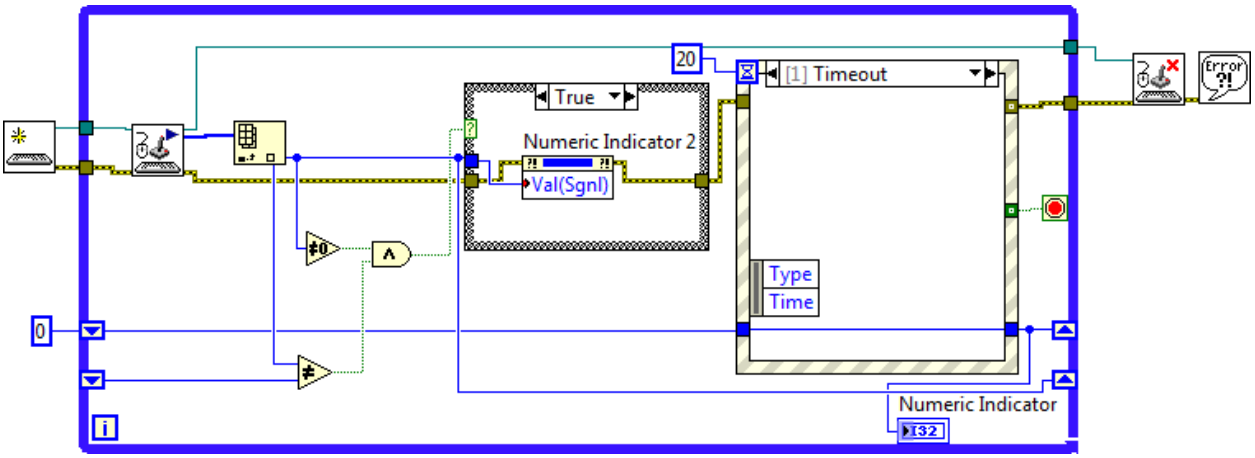

Value Signaling
TF
Value
TF
Numeric Contro
DBL
i
[0] "stop": Value Change
Source
Type
stop
TF

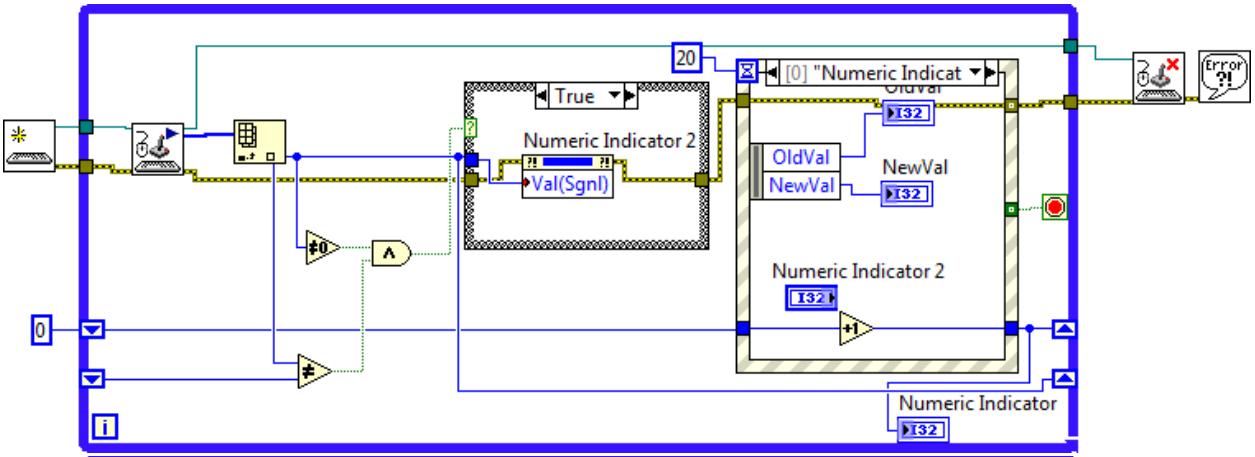

Value Signaling
TF
Value
TF
Numeric Contro
DBL
i
[1] "Value Signaling": Value Chang
Source
Type
Numeric Control
Value (Signaling) Control
DBL
Val(Sgnl)

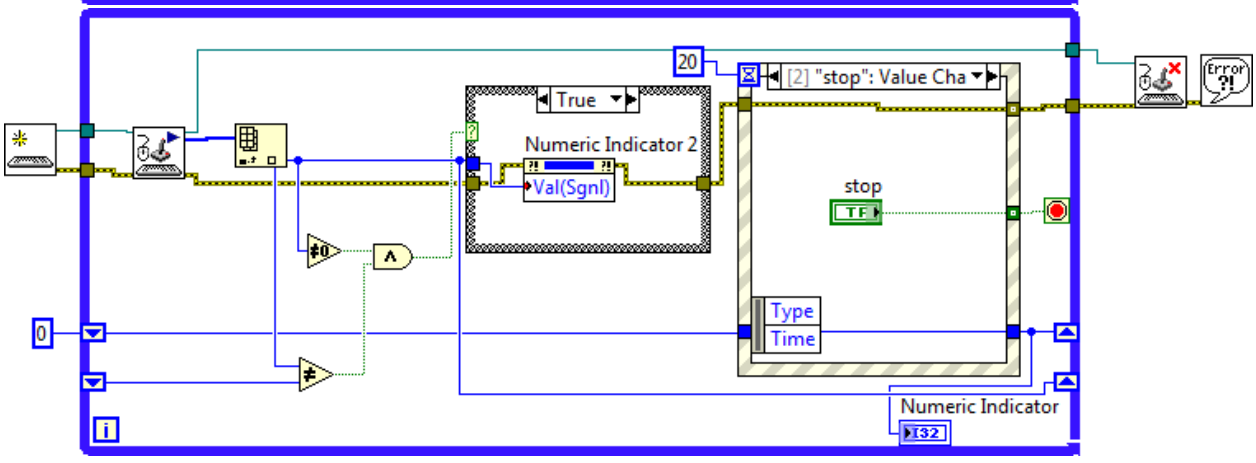

Value Signaling
TF
Value
TF
Numeric Contro
DBL
i
[2] "Value": Value Change
Source
Type
Numer
Value Control
DBL
Value

مثال : برنامه ی زیر را بررسی کنید :









توجه : به روی دکمه های کیبرد فشار دهید آیا تغییری که مشاهده میشود قابل پیش بینی بود ؟

تمرین : برنامه را برای دریافت حروف تغییر دهید.