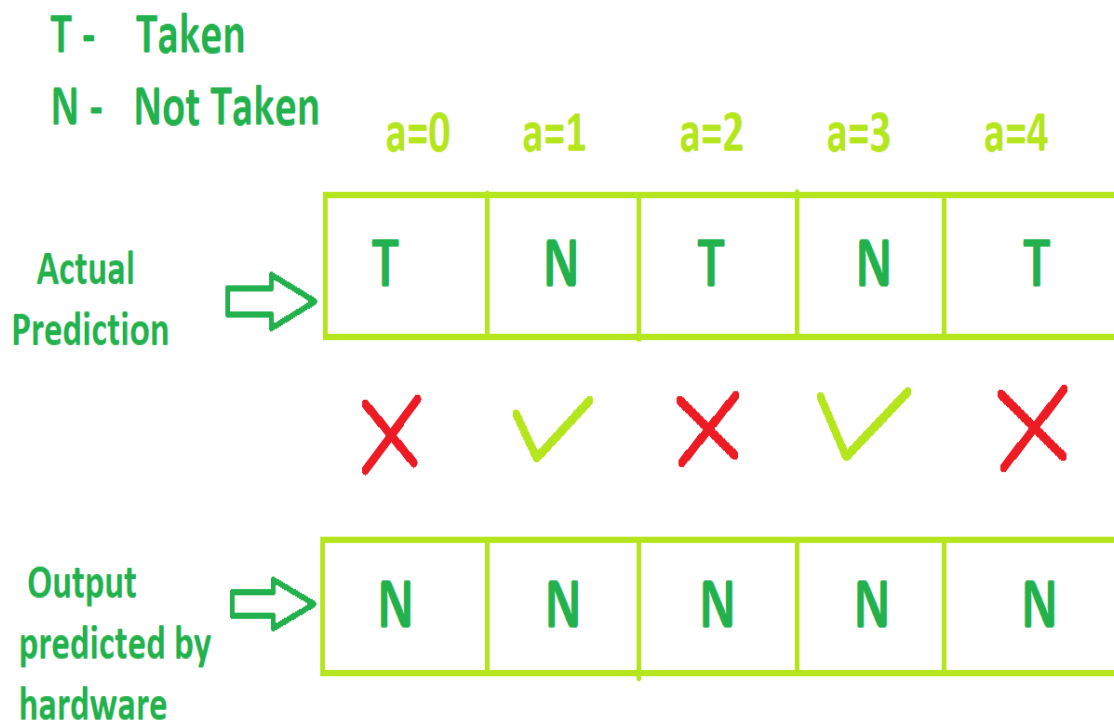


**Static prediction:**

In case of Static branch prediction technique underlying hardware assumes that either the branch is not taken always or the branch is taken always.

**Eg:-**

underlying hardware has assumed that branch is not taken always. The output predicted by underlying hardware and actual output is shown in fig:

**RESULTS:**

ALWAYS TAKEN: - (SCREENSHOTS AFTER RUNNING .PY ON SPYDER(ANACONDA) COMPILER)

```
Console 1/A X
```

```
In [5]: runfile('C:/Users/chaith/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1/STATIC
TAKEN.py', wdir='C:/Users/chaith/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1')
trace_01 prediction_accuracy= 87.87242281954258
trace_02 prediction_accuracy= 57.650113602160815
trace_03 prediction_accuracy= 55.86373984972812
trace_04 prediction_accuracy= 94.49183357454702
trace_05 prediction_accuracy= 49.64661591787319
trace_06 prediction_accuracy= 62.95498610123573
trace_07 prediction_accuracy= 38.98718447467992
trace_08 prediction_accuracy= 33.43497548160043
trace_09 prediction_accuracy= 38.62153535316652
trace_10 prediction_accuracy= 37.58913237409199
trace_11 prediction_accuracy= 37.11875618237318
trace_12 prediction_accuracy= 39.120812032411386
trace_13 prediction_accuracy= 38.62079227463901
trace_14 prediction_accuracy= 38.25414443768126
trace_15 prediction_accuracy= 37.861680868197936
trace_16 prediction_accuracy= 38.73723854298342

In [6]:
```

IPython console History

LSP Python: ready conda: base (Python 3.8.5) Line 1, Col 1 ASCII CRLF RW Mem 66%

ALWAYS NOT\_TAKEN: -

```
Console 1/A X
```

```
In [6]: runfile('C:/Users/chaith/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1/STATIC
NOT_TAKEN.py', wdir='C:/Users/chaith/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1')
trace_01 prediction_accuracy= 12.127577180457424
trace_02 prediction_accuracy= 42.349886397839185
trace_03 prediction_accuracy= 44.13626015027188
trace_04 prediction_accuracy= 5.508166425452991
trace_05 prediction_accuracy= 50.35338408212681
trace_06 prediction_accuracy= 37.04501389876426
trace_07 prediction_accuracy= 61.01281552532007
trace_08 prediction_accuracy= 66.56502451839957
trace_09 prediction_accuracy= 61.37846464683348
trace_10 prediction_accuracy= 62.41086762590802
trace_11 prediction_accuracy= 62.88124381762682
trace_12 prediction_accuracy= 60.879187967588614
trace_13 prediction_accuracy= 61.379207725360985
trace_14 prediction_accuracy= 61.74585556231874
trace_15 prediction_accuracy= 62.138319131802064
trace_16 prediction_accuracy= 61.26276145701658

In [7]:
```

IPython console History

LSP Python: ready conda: base (Python 3.8.5) Line 1, Col 1 ASCII CRLF RW Mem 64%

## Dynamic Prediction: -

In Dynamic branch prediction technique prediction by underlying hardware is not fixed, rather it changes dynamically. This technique has high accuracy than static technique.

Some of them I implemented are:

=> Dynamic-BM

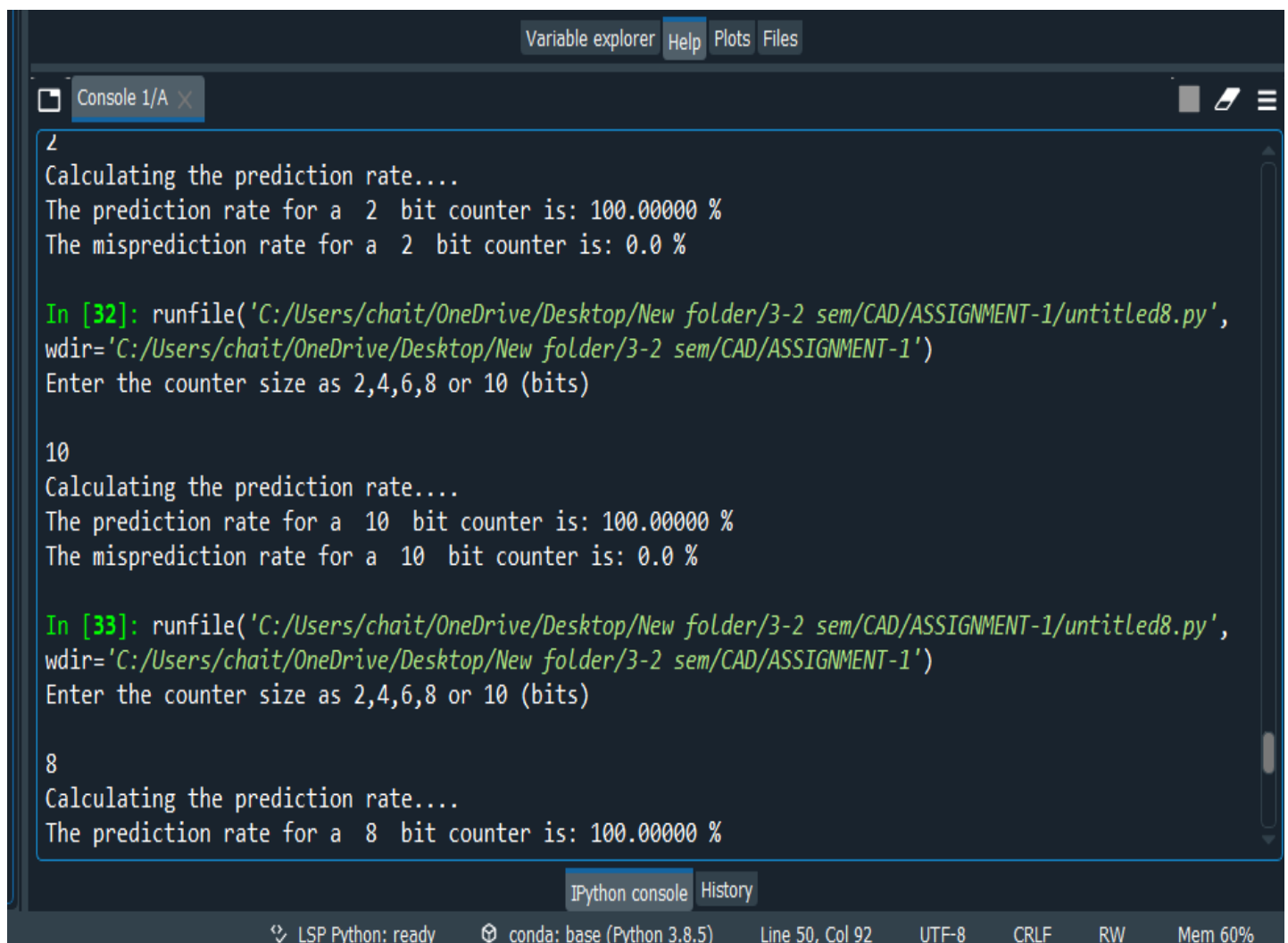
=> Dynamic-GSHARE

=>Local Predictor

## LOCAL PREDICTOR: -

A two-level adaptive predictor remembers the history of the last  $n$  occurrences of the branch and uses one saturating counter for each of the possible  $2^n$  history patterns. ... This means that the last two occurrences of the branch are stored in a two-bit shift register.

(SCREENSHOT OF .PY PROGRAM READING trace\_01 file SHOWING FOR COUNTER SIZES 2,8,10)



```
Variable explorer Help Plots Files
Console 1/A x
2
Calculating the prediction rate....
The prediction rate for a 2 bit counter is: 100.00000 %
The misprediction rate for a 2 bit counter is: 0.0 %

In [32]: runfile('C:/Users/chait/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1/untitled8.py',
wdir='C:/Users/chait/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1')
Enter the counter size as 2,4,6,8 or 10 (bits)

10
Calculating the prediction rate....
The prediction rate for a 10 bit counter is: 100.00000 %
The misprediction rate for a 10 bit counter is: 0.0 %

In [33]: runfile('C:/Users/chait/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1/untitled8.py',
wdir='C:/Users/chait/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1')
Enter the counter size as 2,4,6,8 or 10 (bits)

8
Calculating the prediction rate....
The prediction rate for a 8 bit counter is: 100.00000 %

IPython console History
LSP Python: ready conda: base (Python 3.8.5) Line 50, Col 92 UTF-8 CRLF RW Mem 60%
```

DYNAMIC BI MODEL: (RUN UPTO 9 TRACE FILES )

```
Console 1/A X
In [37]: runtime('C:/Users/chaith/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1/untitled11.py',
wdir='C:/Users/chaith/OneDrive/Desktop/New folder/3-2 sem/CAD/ASSIGNMENT-1')
trace_01 prediction_accuracy= 12.127577180457422 72.76546308274453 72.76546308274453
72.76546308274453 72.76546308274453 72.76546308274453 72.76546308274453
trace_02 prediction_accuracy= 30.571203820959543 183.42722292575726 183.42722292575726
183.42722292575726 183.42722292575726 183.42722292575726 183.42722292575726
trace_03 prediction_accuracy= 37.18098329794176 223.08589978765056 223.08589978765056
223.08589978765056 223.08589978765056 223.08589978765056 223.08589978765056
trace_04 prediction_accuracy= 26.83452596578698 161.00715579472188 161.00715579472188
161.00715579472188 161.00715579472188 161.00715579472188 161.00715579472188
trace_05 prediction_accuracy= 31.720974607165964 190.32584764299577 190.32584764299577
190.32584764299577 190.32584764299577 190.32584764299577 190.32584764299577
trace_06 prediction_accuracy= 32.520424337781165 195.122546026687 195.122546026687 195.122546026687
195.122546026687 195.122546026687 195.122546026687
trace_07 prediction_accuracy= 34.98151335522904 209.88908013137427 209.88908013137427
209.88908013137427 209.88908013137427 209.88908013137427 209.88908013137427
trace_08 prediction_accuracy= 37.508463904104154 225.05078342462494 225.05078342462494
225.05078342462494 225.05078342462494 225.05078342462494 225.05078342462494
trace_09 prediction_accuracy= 39.29479940532154 235.76879643192922 235.76879643192922
235.76879643192922 235.76879643192922 235.76879643192922 235.76879643192922

IPython console History
LSP Python: ready conda: base (Python 3.8.5) Line 32, Col 78 UTF-8 CRLF RW Mem 70%
```