# Take-home Assignment-4

## Advanced Computer Architecture (CS G524)
## Semester-II, 2020-21
## Department of Computer Science and Information Systems (CSIS)
## BITS-Pilani, K K Birla Goa Campus, Goa, India.

Due date: May 12, 2021: 03:00 AM Marks: 65

*Instructions:* Individual student must solve this problem. We can run the tool to measure the similarity of your code with others. Student can get marks zero if the similarity goes beyond 30% and/or evaluators feel that the code is copied from somewhere. The penalty can be applied if the student submits the code beyond the due date and time. A penalty can be measured based on the submission's time.

Create a zip file as <*rollno*>*_simulateBranchPredictor.zip* while submitting it. The file contains all programs and reports. Each file name can have this format: <*rollno*>_*.*. Write the proper name of the variables and use the comment for documentation.

```
int a, b;
a=10000;
while(a>0){ //PC:0b0100_0100
  b=50;
   while (b>0){ //PC: 0b0110_1100
      b=b-2;
   }
   a=a-1;
}
```

**Code Segment-1**

```
int a, b=5;
a=0;
while(a<10000){ //PC:0b0100_0100

   if (a%2==0){ //PC: 0b0110_1100
   }
   a=a+1;
   if (b==0){ //PC: 0b0111_1100
   }
}
```

**Code Segment-2**

```
int a, b, m;
m=10000;
while(m>0){ //PC:0b0100_0100
        a = m%2;
        if (a==2) {//PC: 0b0110_1100
              a=0;
        }
        b=(int) m/2;
        if (b==2) {//PC: 0b0111_0100
              b=0;
        }
         if (a!=b){//PC: 0b1000_1000
          m=m-2;
        }
      m=m-1
}
```

**Code Segment-3**

1. Consider the above code segments and profile their branch outcome. For the code segment 1, 2 and 3, the branch outcome can be stored in the files: branchOutcome1.txt, branchOutcome2.txt and branchOutcome3.txt. For the taken branch outcome stores **T**. Otherwise, stores **NT**. Put ',' between two stored values. Write CPP programs for the above code segments and hold the branch outcome in the given files.

   (Marks: 15)

2. Write a CPP to simulate the generalized *n*-bits saturation branch predictor's behaviour. The branch predictor can read the actual branch outcome from the files: branchOutcome1.txt, branchOutcome2.txt and branchOutcome3.txt. Display the branch prediction accuracy for the *n*=1, 2 and 3 with the following starting states:

   a. All 0s.
   b. All 1s.
   c. A random number generator decides.

   Write a report (.pdf) for the observation and conclusion. (Marks: 25)

3. Write a CPP to simulate the generalized tournament branch predictor's behaviour, (*m* [local history], *n* [local prediction]), (*p* [global history], *q* [global prediction]) and choice prediction (*r*). The branch predictor can read the actual branch outcome from the files: branchOutcome1.txt, branchOutcome2.txt and branchOutcome3.txt. Display the branch prediction accuracy for *m*=3, *n*=2, *p*=3, 4 and 5, *q*=2 and *r*=2 with the following starting state:

   a. All 0s.
   b. All 1s.

   In the local history table (LHT), *m*-bits are used to maintain the history and LHT's locations are decided by the index bits used from the PC. Here, PC's last 3-bits from LSB are used for such indexing. The entries in the LHT is $2^3$.

   In the LPT, *n*-bits are used for the saturation branch predictors. The LHT's *m*-bits history indexes the predictor in the local predictor table (LPT). The entries in the LPT is $2^m$.

   In the global history register (GHR), *p*-bits are used for maintaining the history and which is used for indexing the choice predictor table's (CPT) entry and the global predictor table's (GPT) entry. The number of entries (or rows) depend on the *p*-bits ($2^p$).

   In the GPT, *q*-bits are used for the saturation branch predictors.
   In the CPT, *r*-bits have used the counters.

   *Note*: one can use the previously designed n-bits branch predictor.

   Write a report (.pdf) for the observation and conclusion. (Marks: 25)