

# ECE 586 COMPUTER ARCHITECTURE

## FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

### ALPHA Tournament Predictor:

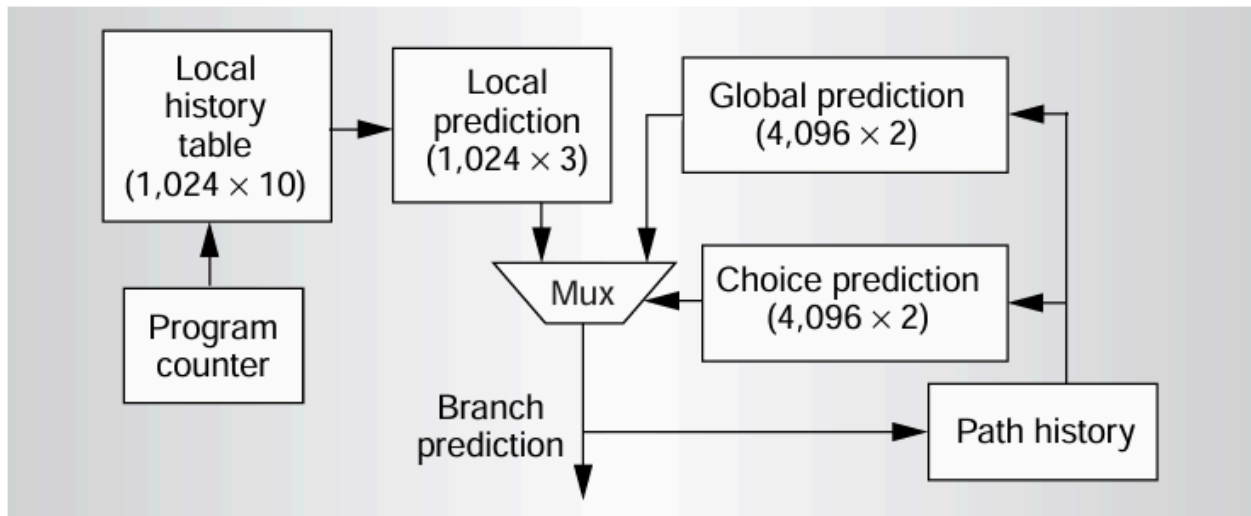
#### Data Structures or Tables:

- Local History Table (1024x10)
- Local Prediction Table (1024x3)
- Global Prediction Table (4096x2)
- Choice Prediction Table (4096x2)
- Global History Bits(path history) (12)

#### Storage:

- $1024 \times 10 + 1024 \times 3 + 4096 \times 2 + 4096 \times 2 + 12 = 29708/8 = 3713.5 \text{ bytes} = 3.7135 \text{ KB}$

#### Block Diagram:



#### Initialization:

- The local predictor table and global predictor are initialized to 0 which is considered as strongly not taken.
- The choice predictor table is initialized to 0.
- Path history is initialized to 0.
- Local history table values are indexed to 0.

#### Algorithm:

##### Indexing and Prediction:

- The local history table is indexed by PC bits[2:11], each entry consists of 10 history bits of a particular branch.
- The local predictor table is indexed by 10 local history bits, predicts branch taken if value is greater than 3 else not taken
- The global predictor and choice predictor tables are indexed by global history.
- The global predictor predicts branches taken if the value is greater than 1 else not taken.
- The choice predictor selects the local predictor's result if values are 0,1 else global predictor's result.
- If the branch is not conditional, we assume prediction as always taken.

## **ECE 586 COMPUTER ARCHITECTURE**

### **FINAL PROJECT REPORT**

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

#### **Updation:**

- If the branch is conditional then comparing if global is predicted true and local prediction is false and choice prediction value is less than 3 increment the choice prediction value of that index.
- If the branch is conditional then comparing if global is predicted false and local prediction is true and choice prediction value is greater than 3.
- Similarly, we update the local predictor table and global predictor table for that branch.
- And we will update the local history table and global history with the outcome.
- If the branch is not conditional then we just update the global history with the branch outcome. As, updating it to the local prediction table increases complexity and misprediction rate.

#### **Competitive Predictors:**

##### **Overview:**

In this project we have implemented three competitive predictors considering different scenarios and references. We have implemented two hybrid predictors and one perceptron based predictor.

#### **Hybrid Predictor :**

##### **Data Structures or Tables:**

- Local History Table (1024x10)
- Local Prediction Table (1024x3)
- Global Prediction Table (4096x2)
- Choice Prediction Table (4096x2)
- Global History Bits(path history) (12)

#### **Storage:**

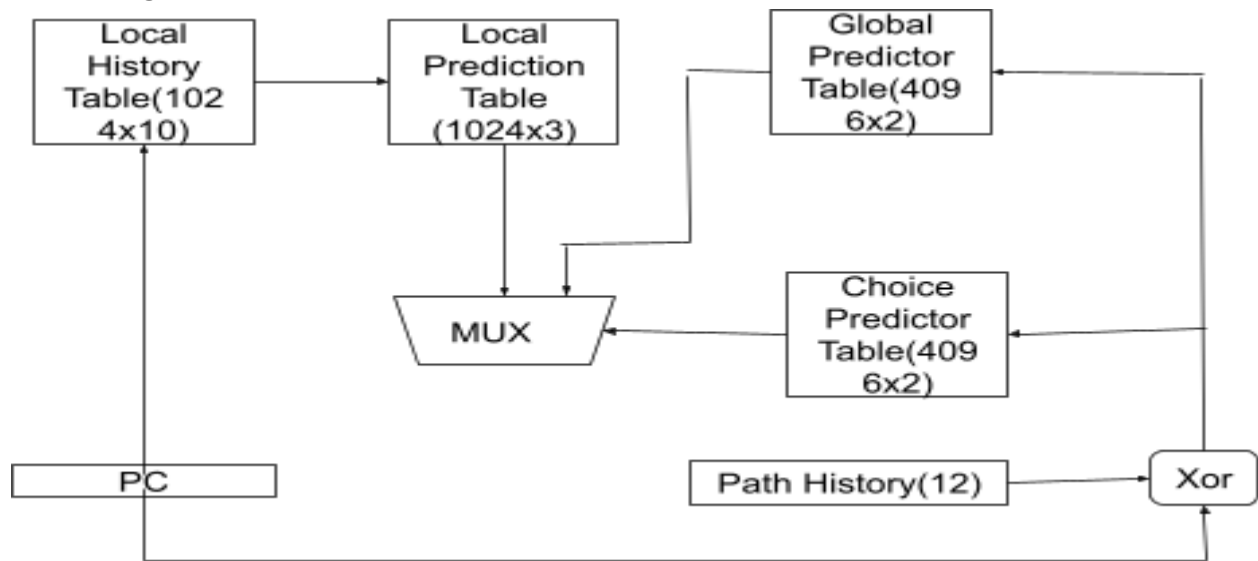
- $1024*10+1024*3+4096*2+4096*2+12= 29708/8 =3713.5 \text{ bytes}= 3.7135 \text{ KB}$

## ECE 586 COMPUTER ARCHITECTURE

### FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

#### Block Diagram:



#### Initialization:

- The local predictor table and global predictor are initialized to 0 which is considered as strongly not taken.
- The choice predictor table is initialized to 0.
- Path history is initialized to 0.
- Local history table values are indexed to 0.

#### Algorithm:

This predictor is similar to the Alpha Predictor.

#### Indexing and Prediction:

- We have considered using the index of the global prediction table and choice predictor as hashing of PC index and path history bits.
- $global\_idx = PCbits \text{ xor } path\_historybits$
- The local history table is indexed by PC bits[2:11], each entry consists of 10 history bits of a particular branch.
- The local predictor table is indexed by 10 local history bits, predicts branch taken if value is greater than 3 else not taken
- The choice predictor indexed by hashed result chooses between the local predictor and global predictor. If the value is 0,1 local predictor result is given out else global predictor result.
- The global predictor indexed by hashed result predicts branch taken if the value is greater than 1 else not taken.
- Used 2-bit Saturation counters in the global Predictor table and choice predictor table
- Used 3-bit Saturation counter in Local Predictor Table.

# **ECE 586 COMPUTER ARCHITECTURE**

## **FINAL PROJECT REPORT**

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

### **Updation:**

- If the branch is conditional then comparing if global is predicted true and local prediction is false and choice prediction value is less than 3 increment the choice prediction value of that index.
- If the branch is conditional then comparing if global is predicted false and local prediction is true and choice prediction value is greater than 3.
- Similarly, we update the local predictor table and global predictor table for that branch.
- And we will update the local history table and global history with the outcome.
- If the branch is not conditional then we just update the global history with the branch outcome. As, updating it to the local prediction table increases complexity and misprediction rate.

### **Conclusions:**

- This Hybrid Predictor outperforms the tournament predictor in all standards INT, MM, FP, SERV.
- As per the statistics mentioned below in table 1, the misprediction rate of hybrid predictor is slightly lesser than the tournament predictor.

### **References:**

- This logic is inferred from the Tournament predictor, G-Share Predictor and tage hybrid predictor.
- Lecture Slides

### **Perceptron Predictor:**

Constraints Given : storage budget is no more than 4k+4 bytes

Given a fixed hardware budget, three parameters need to be tuned to achieve the best performance: the history length, the number of bits used to represent the weights, and the threshold.

### **Data Structures or tables:**

- Global History register: 28 //best history range as per the paper is 12 to 62
- As per the paper the best history length for 4kB storage is 28.
- Perceptron table:128 entries // as there is a feature mentioned table size should be in powers of 2.
- Bits in weight : 8

### **Storage:**

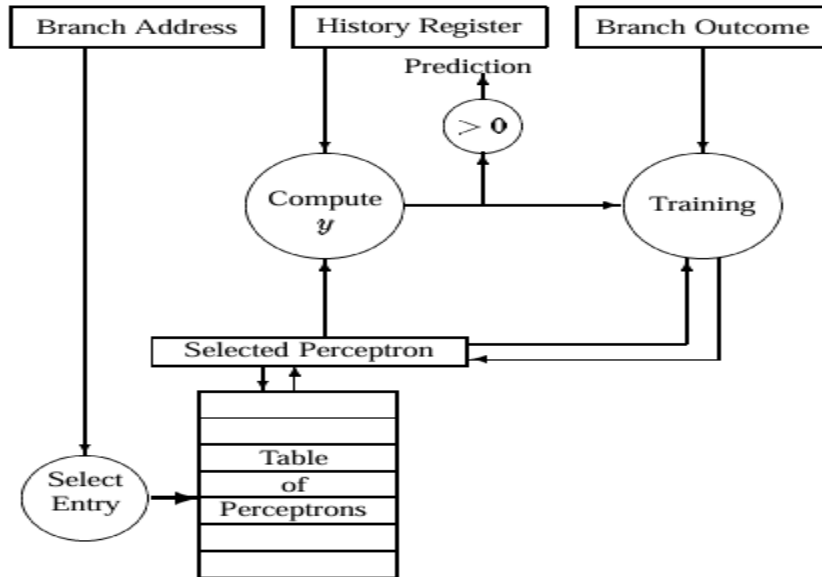
Size of predictor =  $128 * 8 * (28 + 1) = 29,696 \text{ bits} = 3.712 \text{ KB}$

## ECE 586 COMPUTER ARCHITECTURE

### FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

#### Block Diagram:



#### Initialization:

- The Table of perceptrons is initialized to zero.
- The global history register is initialized to zero.
- The number of weights is considered as 8.
- Bits for the weight are 8.
- Considering budget as

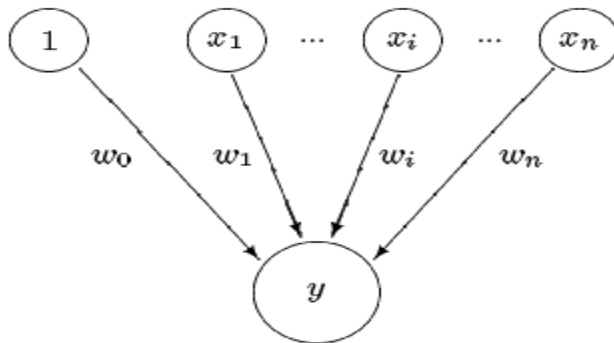
#### Algorithm:

- As cited in the paper the best threshold for given history length is :  $1.93 \cdot \text{hist\_len} + 14$ .
- The branch address is hashed to produce an index  $i$   $0::N-1$  into the table of perceptrons.
- The  $i$ th perceptron is fetched from the table into a vector register,  $P[0::n]$ , of weights.
- (The Perceptron predictor finds an individual correlation factor for each bit position in the global history)
- The value of  $y$  is computed as the dot product of  $P$  and the global history register.
- The branch is predicted not taken when  $y$  is Negative, or taken otherwise.
- Once The Actual Outcome Of The Branch Becomes Known, the training algorithm uses this outcome and the value of  $y$  to update the weights in  $P$ .
- $P$  is written back to the  $i$ th entry in the table.

## ECE 586 COMPUTER ARCHITECTURE

### FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham



shows graphical model of a perceptron

$$y = w_0 + \sum_{i=1}^n x_i w_i.$$

Updating weights:

```
if sign(yout) ≠ t or |yout| ≤ θ then
    for i := 0 to n do
        wi = wi + txi
    end for
end if
```

#### Conclusions:

- Unfortunately it could not perform well over tournament predictor or perceptron predictor in all the standards. This predictor performs the worst in the SERV standard.
- As it considers only a single layer predictor this predictor does not perform well over two layer predictors like tournament predictor.

#### References:

- Dynamic Branch Prediction with Perceptrons by Daniel A. Jimenez, Calvin Lin
- Link: <https://www.cs.utexas.edu/~lin/papers/hpca01.pdf>

#### Hybrid Predictor 1:

##### Data Structures or Tables:

- Global Prediction Table(8192x2)
- Local Prediction Table(4096x2)
- Choice Prediction Table(4096x2)
- Path history(13)

##### Storage:

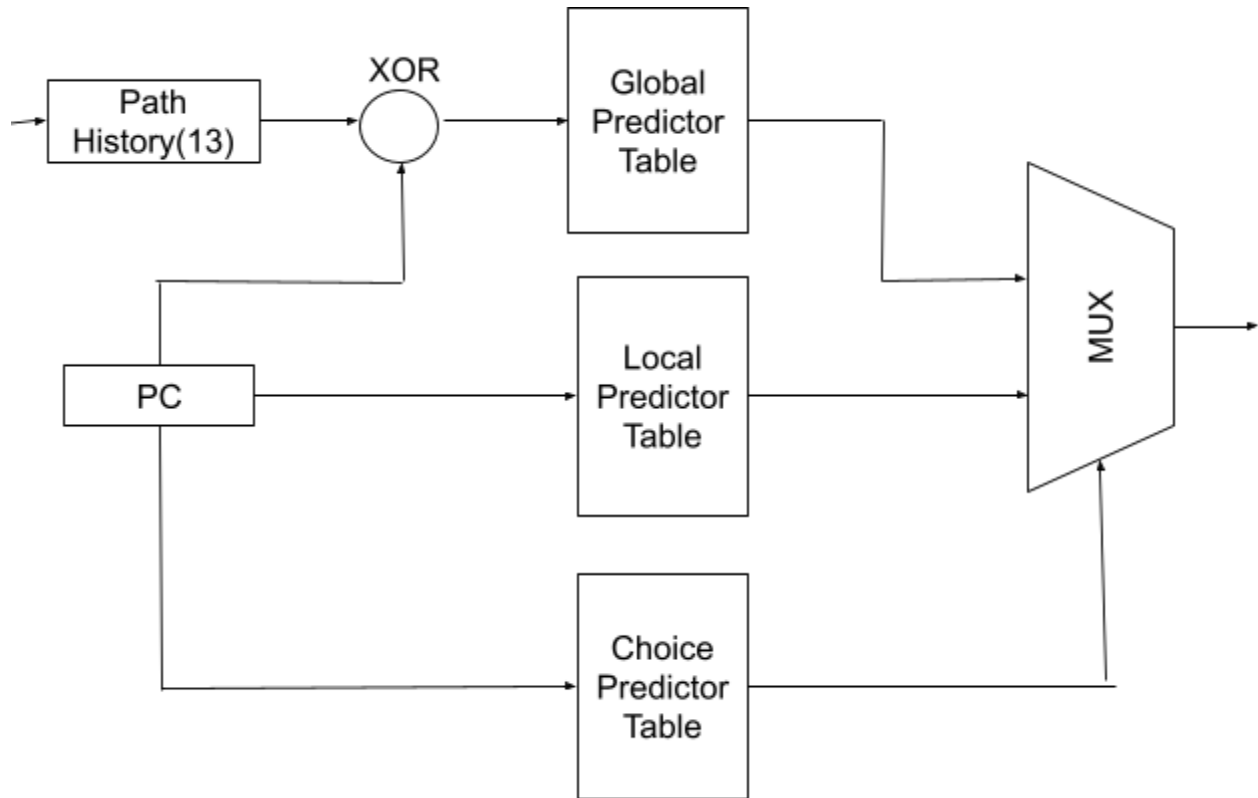
- Size of predictor = 8192\*2+4096\*2+4096\*2+13=32781 = 4.09KB

## ECE 586 COMPUTER ARCHITECTURE

### FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

#### Block Diagram:



#### Algorithm:

##### Indexing:

- In this, path history bits and PC 13 bits are xored to index the global predictor table.
- Local predictor table and choice predictor table is indexed by 12 PC bits.

##### Prediction:

- In the global predictor table and local predictor table, we are using a 2-bit saturating counter, the prediction is true if the value is greater than 1 else false.
- In Choice predictor, we used 2-bit saturating counter, where we choose local prediction if value is less than 2 else global prediction.

##### Updation:

- The path history or global history is updated using the prediction outcome.
- The prediction tables are updated similarly, depending on the value in the predictor tables and features of the saturating counter.

#### References:

- Analysis and Optimization of the Branch Prediction Unit of SweRV EH1
- Link: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9996038>

## ECE 586 COMPUTER ARCHITECTURE

### FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

#### Experimental Results:

The table represents the mispredicts / 1000 instructions of different predictors for various standards.

Standards	Tournament Predictor	Hybrid Predictor 0	Perceptron Predictor	G-SHARE Predictor	Hybrid Predictor 1
INT-1	7.397	7.273(o)	7.964	8.593	8.829
INT-2	9.715	9.597(o)	11.175	9.105	8.834
INT-3	12.050	12.756	11.864(o)	15.191	13.728
INT-4	2.425	2.286(o)	3.252	2.624	2.778
INT-5	0.406	0.394	0.378(o)	0.565	0.496
MM-1	8.299	7.991	7.572(o)	8.556	8.639
MM-2	10.970	10.766	9.792(o)	11.701	10.707
MM-3	2.021	1.800(o)	3.785	5.537	5.480
MM-4	2.165	2.034	1.618(o)	2.135	2.059
MM-5	6.436	6.115	7.753	7.210	5.841(o)
FP-1	3.286	3.080	2.497(o)	4.092	4.554
FP-2	1.317	1.135	1.101(o)	1.217	1.208
FP-3	0.518	0.473(o)	0.528	0.622	0.514
FP-4	0.266	0.265	0.210(o)	0.319	0.303
FP-5	1.397	0.336(o)	0.791	1.856	1.698
SERV-1	9.853	9.693	17.844	7.247	6.094(o)
SERV-2	10.299	10.228	18.775	7.678	6.359(o)
SERV-3	7.687	7.512	11.117	7.580	6.349(o)
SERV-4	9.492	9.537	14.742	7.450	6.238(o)
SERV-5	9.780	9.755	15.759	7.781	6.365(o)

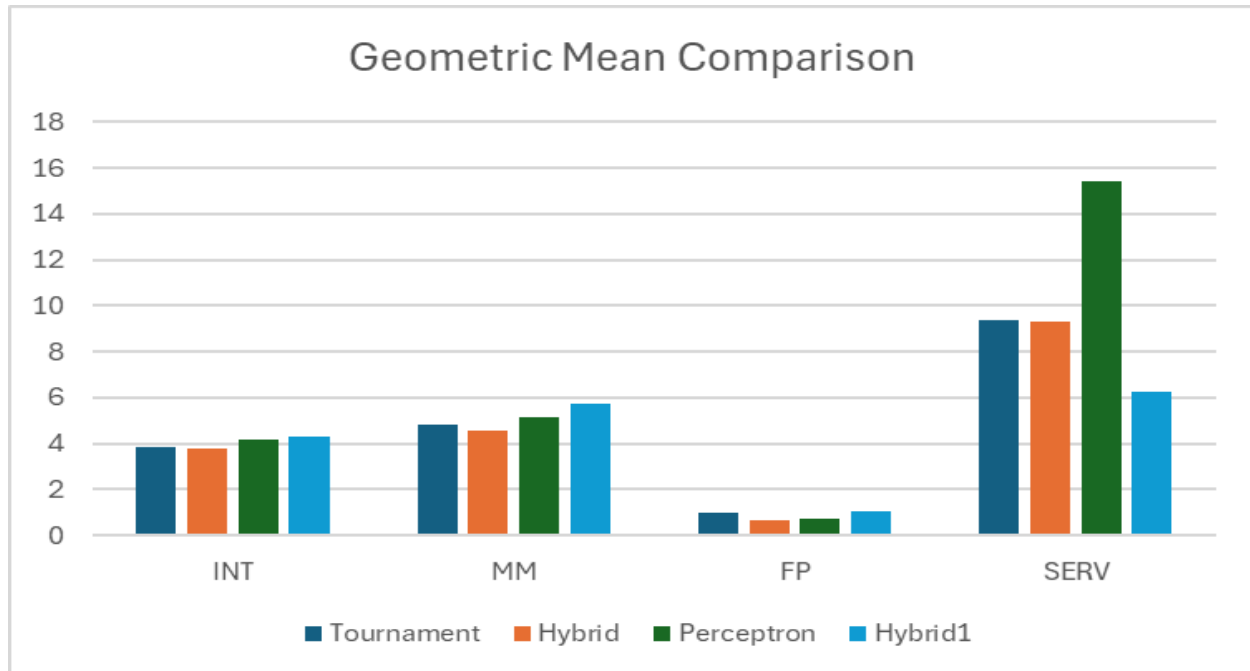


## ECE 586 COMPUTER ARCHITECTURE

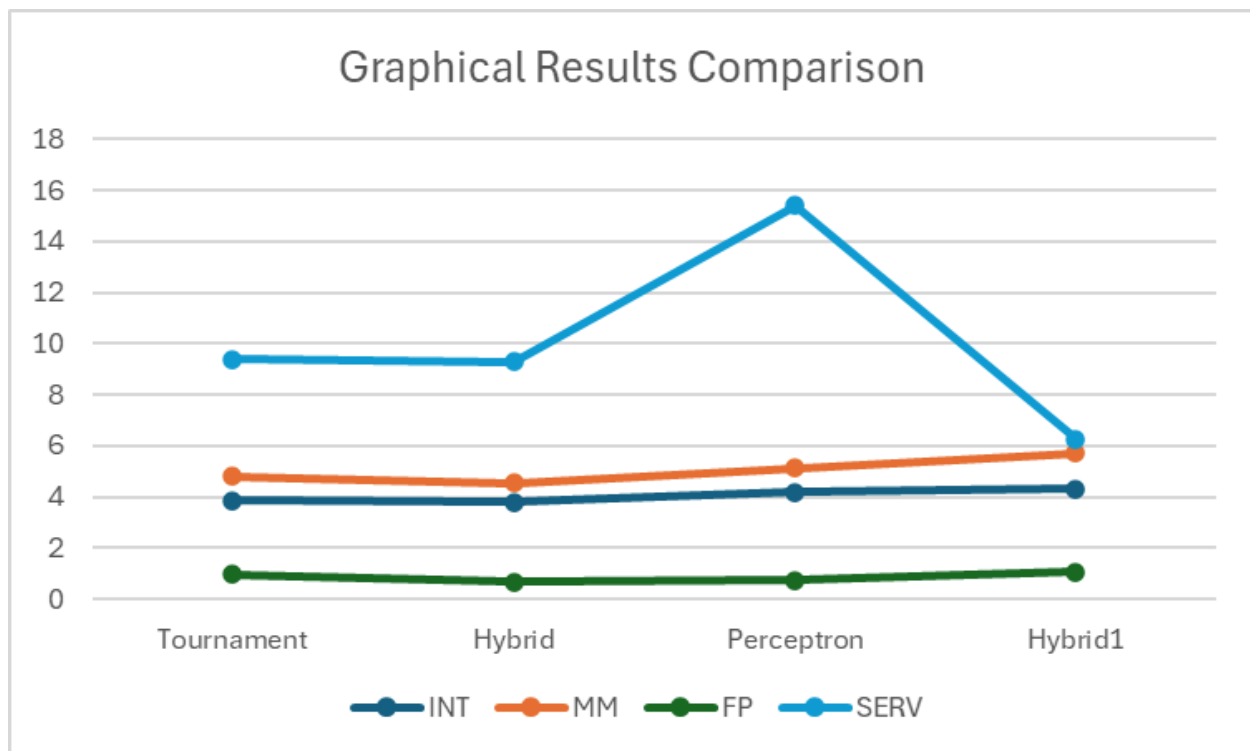
### FINAL PROJECT REPORT

Team 16 | Sai Kumar Reddy Pulagam | Reshwanth Sri Sai Sesham

(o)-represents which predictor performs best in particular standard traces



Bar Graph represents the geometric means for different predictors on INT, MM, FP, SERV standards



Plotting represents the geometric means for different predictors on INT, MM, FP, SERV standards