

1. **Visualizing analysis in a GUI (5 points):** DrCCTProf can generate output in a specific format that can be visualized using DrCCTProf View extension in VSCode. You need to implement this and use VSCode to visualize the results. Besides the client file, you also need to give a screen snapshot of the VSCode that visualizes the analysis result. The details about how to support VSCode and visualize it will be added soon.

*Ans: Screenshot has been attached with the submission*

2. **Using the overflow bit in the state register to detect (5 points) and compare (5 points) it with the approach implemented in this project (i.e., the algorithm in Slide 12).** For the detection, you need to implement it. For the comparison, you need to provide a document to compare the results for different test cases you designed. A starting point for testing is designing overflows with both signed and unsigned integers to check which solution is better. Moreover, in the document, you should describe how you implement the overflow bit solution.

*Ans: The overflow bit's position is in the 11th. So the bitmask = (0x0800). We need to simply do a logical bitwise operation here.*

```
If (bitFlagsValue & bitmask) {  
  
    // the overflow bit is 1  
  
}  
  
else {  
  
    // the overflow bit is 0  
  
}
```

*If we check the overflow by the means of the overflow bit of the flag register, it will be more efficient by the means of cost. It will be less expensive as the comparison cost will be less.*

3. **You can describe your own thoughts about improving the detection of integer overflows. Please document your thoughts and try to implement them. We will give extra credits (up to 10 points) according to your efforts and insights.**

*Ans: I think the improvement of detection of integer overflow lies in the cross relationship of different flag bits of flag register. As we know the sign bit, overflow flag bit, carry bit, borrow bit - are closely related. And we can develop a good relation between these flag bits which can leverage our integer overflow detection within less cost and more effective manner.*