# EE/CprE/SE 492 - sdmay21-10
# Dynamic Programming Based FPGA Hosted Control-Flow Integrity (CFI)
# Report 5

*March 15th - March 29th*
*Client: Akhilesh Tyagi*
*Faculty Advisors: Zelong Li, Ananda Biswas*

## Team Members

Gregory Wendt - Meeting Scribe
Cole Schumacher - Meeting Facilitator
Nickolas Mitchell - Chief Engineer (FPGA)
Sam Henley - Chief Engineer (Software)
Maxwell Wangler - Test Engineer
Tristan Duyvejonck - Report Manager

---

## Past Weeks Accomplishments

Over the past two weeks, our team has finished our C code to the extent that it matches the current Java code. Furthermore, Nick completed the VHDL implementation of our race logic algorithm and started looking into ways to communicate between a computer and the FPGA. Lastly, Sam continued researching and developing the LLVM pass required for our project.

## Pending Issues

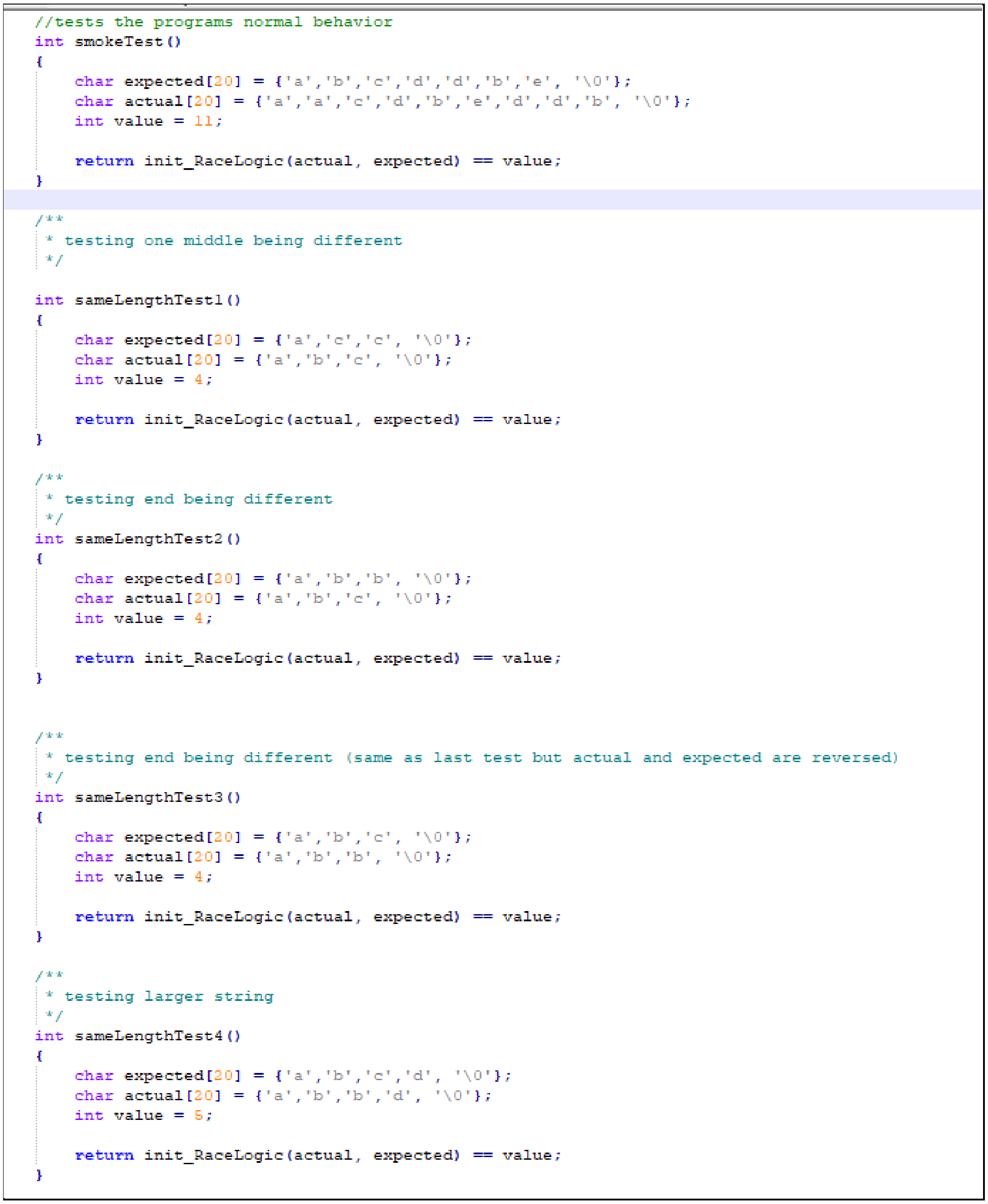
- Work towards a better understanding of LLVM compilers
  - Develop pass for inserting a function after each basic block
- Test VHDL code
- Brainstorm ways to send data to the FPGA board.
  - FPGA solution will need two string inputs for the program to use
  - Could use meta data from a header file or research other methods.
  - Output from the FPGA program then needs to be sent back to the computer to be analyzed.

## Individual contributions

| NAME | Individual Contributions<br>(Quick list of contributions. This should be short.) | Hours this week | HOURS cumulative |
|---|---|---|---|
| Gregory Wendt | Finished C tests so that it matches the Java code. Added information about possible future features to the design document | 3 | 17 |
| Cole Schumacher | Helping coordinate meetings with TA's and team, Research and development into FPGA and implementation with Quartus Prime/Modelsim | 4 | 16 |
| Nickolas Mitchell | Completed VHDL code/started researching how to communicate between computer and FPGA. | 9 | 41 |
| Sam Henley | Continued research and development with LLVM | 4 | 23 |
| Maxwell Wangler | Finished the C code, and finished the test suite for the C code. | 2 | 14.5 |
| Tristan Duyvejonck | Was not able to contribute | 0 | 3 |

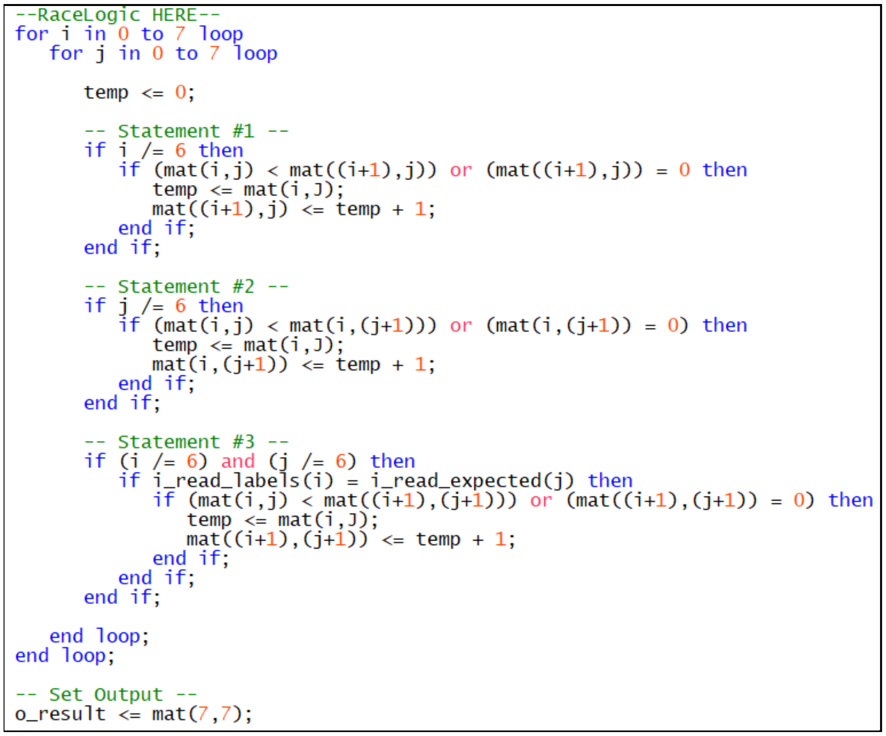
## Plans for the Upcoming Report

- Gregory
    - Communication between FPGA and computer.
- Cole
    - Research simulation tools in Quartus Prime
- Nickolas
    - Test VHDL code
    - Continue researching ways to communicate with FPGA from computer
        - Most promising is UART as of right now
- Sam
    - Complete function insert pass
- Maxwell
    - Figure out communication between FPGA board and the computer.
- Tristan
    - Work with Sam developing LLVM passes



```c
//tests the programs normal behavior
int smokeTest()
{
    char expected[20] = {'a','b','c','d','d','b','e', '\0'};
    char actual[20] = {'a','a','c','d','b','e','d','d','b', '\0'};
    int value = 11;

    return init_RaceLogic(actual, expected) == value;
}

/**
 * testing one middle being different
 */

int sameLengthTest1()
{
    char expected[20] = {'a','c','c', '\0'};
    char actual[20] = {'a','b','c', '\0'};
    int value = 4;

    return init_RaceLogic(actual, expected) == value;
}

/**
 * testing end being different
 */
int sameLengthTest2()
{
    char expected[20] = {'a','b','b', '\0'};
    char actual[20] = {'a','b','c', '\0'};
    int value = 4;

    return init_RaceLogic(actual, expected) == value;
}


/**
 * testing end being different (same as last test but actual and expected are reversed)
 */
int sameLengthTest3()
{
    char expected[20] = {'a','b','c', '\0'};
    char actual[20] = {'a','b','b', '\0'};
    int value = 4;

    return init_RaceLogic(actual, expected) == value;
}

/**
 * testing larger string
 */
int sameLengthTest4()
{
    char expected[20] = {'a','b','c','d', '\0'};
    char actual[20] = {'a','b','b','d', '\0'};
    int value = 5;

    return init_RaceLogic(actual, expected) == value;
}
```

Figure 1.1 Sample of test in C code



```
--RaceLogic HERE--
for i in 0 to 7 loop
   for j in 0 to 7 loop

      temp <= 0;

      -- Statement #1 --
      if i /= 6 then
         if (mat(i,j) < mat((i+1),j)) or (mat((i+1),j)) = 0 then
            temp <= mat(i,J);
            mat((i+1),j) <= temp + 1;
         end if;
      end if;

      -- Statement #2 --
      if j /= 6 then
         if (mat(i,j) < mat(i,(j+1))) or (mat(i,(j+1)) = 0) then
            temp <= mat(i,J);
            mat(i,(j+1)) <= temp + 1;
         end if;
      end if;

      -- Statement #3 --
      if (i /= 6) and (j /= 6) then
         if i_read_labels(i) = i_read_expected(j) then
            if (mat(i,j) < mat((i+1),(j+1))) or (mat((i+1),(j+1)) = 0) then
               temp <= mat(i,J);
               mat((i+1),(j+1)) <= temp + 1;
            end if;
         end if;
      end if;

   end loop;
end loop;

-- Set Output --
o_result <= mat(7,7);
```

Figure 2.1 VHDL RaceLogic