High Level Synthesis

Homework 3

Konstantinos Roumoglou

May 2, 2024

1 EXERCISE 1

- We start our implementation in Catapult and we define as top module to hierarchy the function **"compute_row_sum"**.
- We select the appropriate libraries. In mapping,we set the clock at 500 MHz.
- Then in architecture part we define memory interfaces for input variables as: **"ccs_sample_mem.css_ram_sync_1R1W"**.
- We run simulation(RTL) and table results are the following.

compute_row_sum<3>.v1 (extract)	31	62.00	38	76.00	202.20	0.28

Figure 1: Table

-
- We try to insert pipelines and fail. This is because there is dependency within the for loops due to row_sum[i] += a[i][j]. The calculation of the sum of each row depends on the previous results.
- We remake our code to be able to insert pipeline.

```
1  void CCS_BLOCK(compute_row_sum)(short a[N][N], short row_sum[N]) {
2    short sum;
3    ROW: for (int i=0; i < N; i++) {
4      sum=0;
5      // row_sum[i] = 0;
6      SUM: for (int j=0; j <N; j++) {
7      //   row_sum[i] += a[i][j];
8      sum+=a[i][j];
9      }
10     row_sum[i]=sum;
11   }
12 }
```

- After that change we run the simulation and we take the following results from Table.

compute_row_sum<3>.v1 (extract)	9	18.00	13	26.00	354.95	0.23

- We observe that runtime is significantly reduced.
- Finally,we confirm the correct operation of the code via Questa Sim.

```
Info: Collecting data completed
   captured 100 values of a
   captured 100 values of row_sum
Info: scverify_top/user_tb: Simulation completed

Checking results
'row_sum'
   capture count        = 100
   comparison count     = 100
   ignore count         = 0
   error count          = 0
   stuck in dut fifo    = 0
   stuck in golden fifo = 0

Info: scverify_top/user_tb: Simulation PASSED @ 2604 ns
** Note: (vsim-6574) SystemC simulation stopped by user.
```

2 EXERCISE 2

- We define again the same parameters for Catapult.
- We run the code and we get the following results from table.

Solution	Latency...	Latency...	Throug...	Throug...	Total Area	Slack
fibonacci<5>.v1 (extract)	18	36.00	22	44.00	624.35	0.66

- We try to insert pipeline to loop **"LOOP"** but it is not possible.
- This happens because calculation of $fib[i]$ presupposes the calculation of two swelling elements, $fib[i-1]$, $fib[i-2]$.So every iteration depends of previous calculations.
- So we remake out function like this :

```
template<int N>
void CCS_BLOCK(fibonacci)(int fib[N]) {

    int a = 1;
    int b = 1;

    fib[0] = a;
    fib[1] = b;

    FIB_LOOP: for (int i = 2; i < N; i++) {
        int c = a + b;
        a = b;
        b = c;
        fib[i] = c;
    }
}
```

- We run simulation and we get the following results:

fibonacci<5>.v10 (extract)	6	12.00	8	16.00	152.00	1.53

- Finally we confirm the correct operation of our circuit via Questa Sim.

```
VSIM 2> run -all
# SCVerify intercepting C++ function 'fibonacci<5>' for RTL block 'fibonacci_5'
# Expected: 1 1 2 3 5
# Actual:   1 1 2 3 5
# CORRECT!
# Info: Execution of user-supplied C++ testbench 'main()' has completed with exit code = 0
#
# Info: Collecting data completed
#    captured 1 values of fib
# Info: HW reset: TLS_rst active @ 0 s
# Info: scverify_top/user_tb: Simulation completed
#
# Checking results
# 'fib'
#    capture count       = 1
#    comparison count    = 1
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
#
# Info: scverify_top/user_tb: Simulation PASSED @ 22 ns
# ** Note: (vsim-6574) SystemC simulation stopped by user.
# 1
#
```

3 EXERCISE 3

- The programm cannot merge two loops because they have different limits and some dependencies at sum's calculation.
- When we run simulation we take the following results from Table.

Solution	Latency...	Latency...	Throug...	Throug...	Total Area	Slack
two_loops.v1 (extract)	220	440.00	222	444.00	923.92	0.65

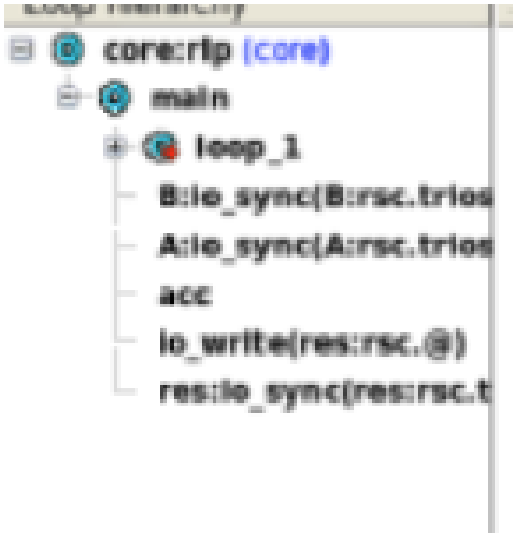
- We remake our code so that the Catapult could merge the two loops.

```
1  void CCS_BLOCK(two_loops)(int A[10], int B[100], int &res) {
2
3   int sum_a = 0;
4   int sum_b=0;
5
6      loop_1: for (int i = 0; i < 100; i++) {
7          if (i < 10) {
8              sum_a += A[i];
9              }
10     }
11
12     loop_2: for (int i=0;i<100; i++){
13
14             sum_b += B[i];
15
16     }
17
18
19     res = sum_a+sum_b;
20 }
```

- We run the code first without pipeline and then within pipeline(II=1).

two_loops.v4 (extract)	301	602.00	303	606.00	1462.73	0.65
two_loops.v5 (extract)	102	204.00	104	208.00	1463.66	0.65

- We observe that the Catapult merge the two loops.



- Finally, we confirm the correct operation of our circuit via Questa Sim.

```
# Expected= 1713424835, Actual= 1713424835. CORRECT!
# Info: Execution of user-supplied C++ testbench 'main()' has completed with exit code = 0
#
# Info: Collecting data completed
#    captured 10 values of A
#    captured 10 values of B
#    captured 10 values of res
# Info: scverify_top/user_tb: Simulation completed
#
# Checking results
# 'res'
#    capture count       = 10
#    comparison count    = 10
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in goldee fifo = 0
#
# Info: scverify_top/user_tb: Simulation PASSED @ 2004 ns
# ** Note: (vsim-6574) SystemC simulation stopped by user.
# 1
#
```