GTU

DEPARTMENT OF COMPUTER ENGINEERING

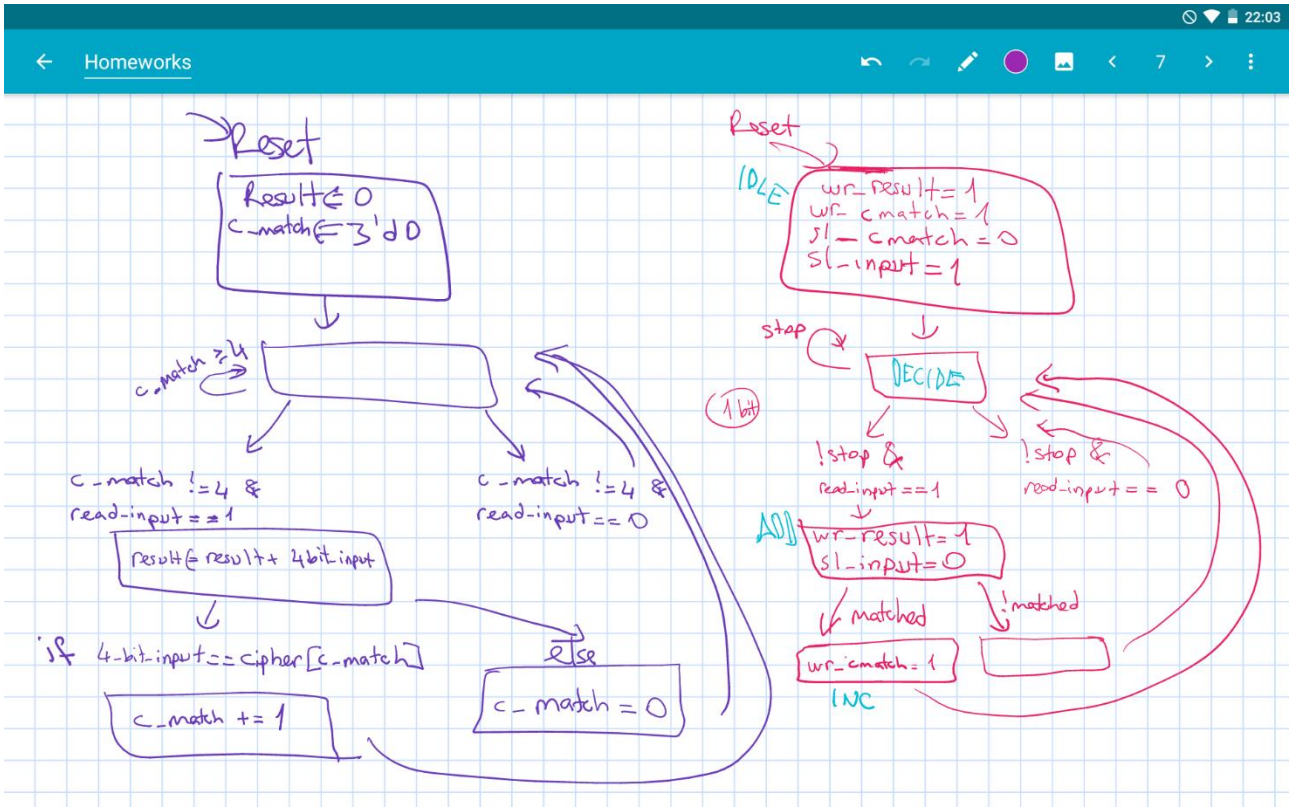
CSE 331 – Autumn 2022

HOMEWORK 4 REPORT

SÜLEYMAN GÖLBOL

1801042656

FSM FOR DATAPATH



While writing the code, I used the multiplier solution of teacher in class as my base file components.

CONTROL UNIT

1.

```
//state register
always @ (posedge clk) begin
    if(reset)
        current_state <= IDLE;
    else
        current_state <= next_state;
end
```

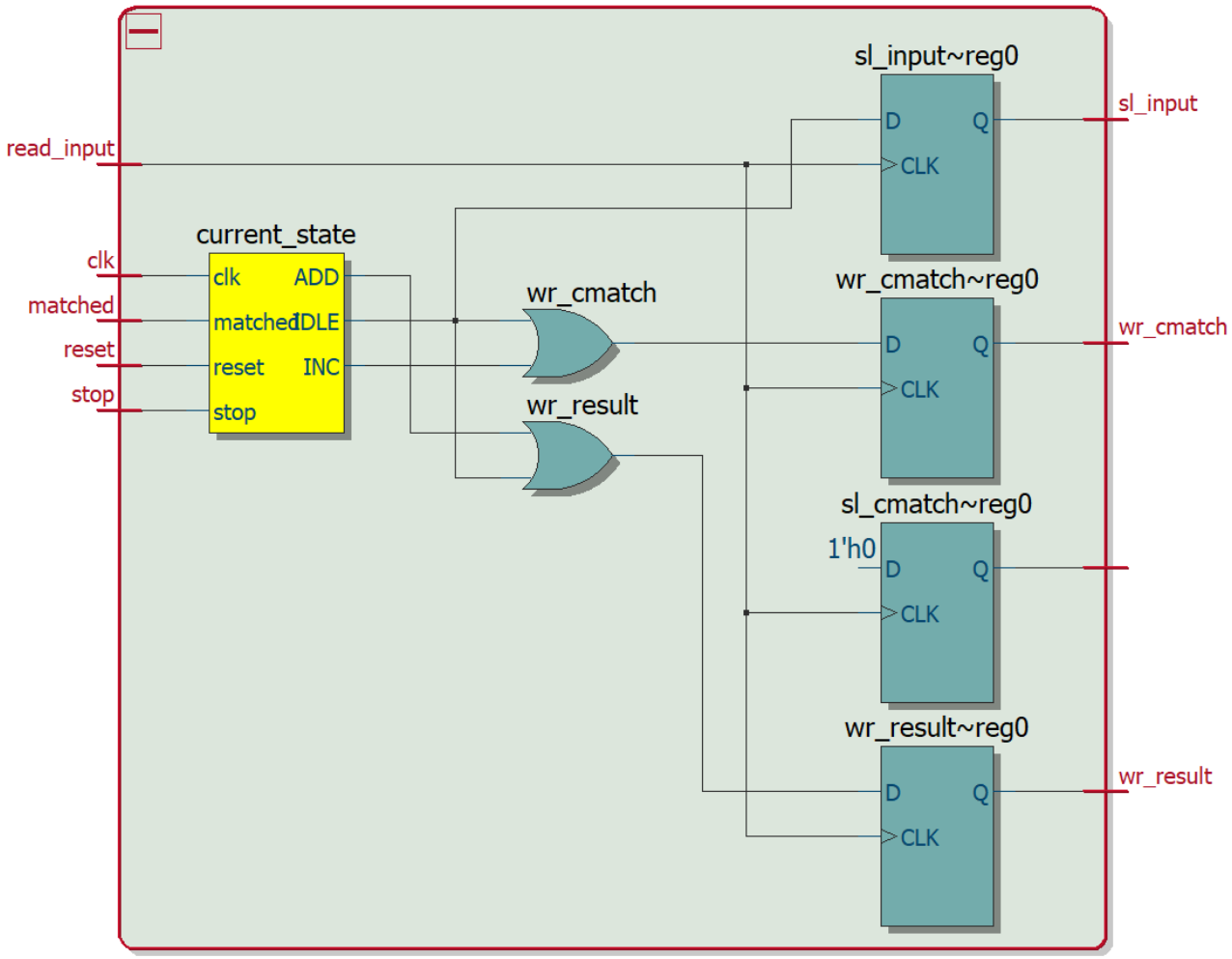
2.

```
always @ (*) begin
   case (current_state)
      IDLE: begin
         next_state = DECIDE;
      end

      DECIDE: begin
         if(stop)
            next_state = DECIDE;
         else
            next_state = ADD;
      end

      ADD: begin
         if(matched)
            next_state = INC;
         else
            next_state = DECIDE;
      end

      INC: begin
         next_state = DECIDE;
      end
   endcase
end
```

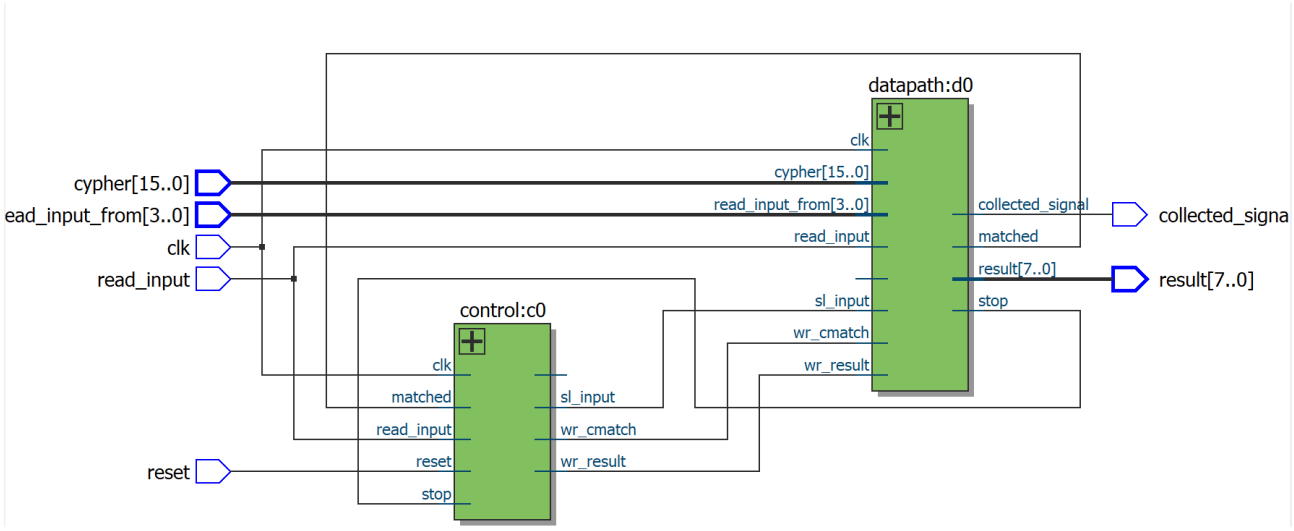
3.

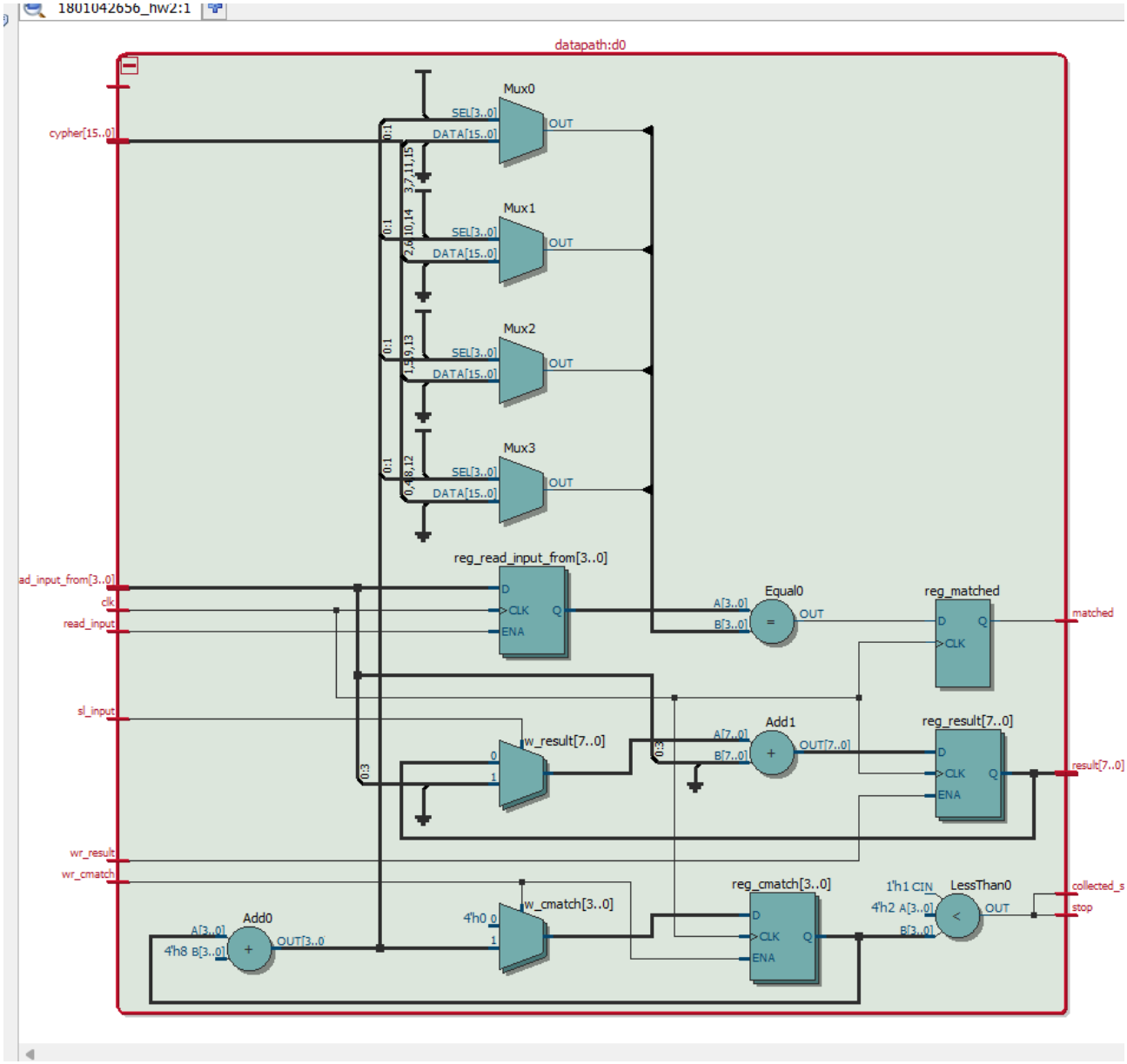
```
always @ (posedge read_input) begin
   wr_result         = 1'b0;
   wr_cmatch         = 1'b0;
   sl_cmatch         = 1'b0;
   sl_input          = 1'b0;
   // matched        = 1'b0;

   case (current_state)
      IDLE: begin
         wr_result         = 1'b1;
         wr_cmatch         = 1'b1;
         sl_cmatch         = 1'b0;
         sl_input          = 1'b1;
      end

      DECIDE: begin
         // no change inside
      end

      ADD: begin
         wr_result      = 1'b1;
         sl_input       = 1'b0;
      end

      INC: begin
         sl_input       = 1'b0;
         //sl_cmatch    = 1'b1;
         wr_cmatch      = 1'b1;
      end
   endcase
end
```

3 always block used as wanted.



DATAPATH DETAILS





RESULTS

When readinput is 0, it doesn't get the input number.

It calculates the sum result correctly but sometimes because of the clock syncing problem, it doesn't sum up some numbers.

Also I couldn't finish the detecting cypher part.

Due to my other course's final exams, I couldn't complete the task.

```
#16
reset = 1'b0;
read_input_from = 4'b0001;

#16
read_input_from = 4'b0010; //2

#16
read_input_from = 4'b0011; //3

#16
read_input_from = 4'b0100; //4

#16
read_input_from = 4'b0101; //5

#16
read_input_from = 4'b0110; //6

#16
read_input_from = 4'b0111; //7

#16
read_input_from = 4'b1000; //8

#16
read_input_from = 4'b0000; // 0
```

Note: it prints summation results multiple times for same thing.

```
# ReadInput: 1, Cypher: 0100001100100001
# Read input from: 0001
# Result:
#                          00000001
#
step -out -current
# Break in Module testbench_cypher_detect at C
step -out -current
# ReadInput: 0, Cypher: 0100001100100001
# Read input from: 0001
# Result:
#                          00000001
#
```

Second

```
step -out -current
# ReadInput: 1, Cypher: 0100001100100001
# Read input from: 0010
# Result:
#                          00000011
#
```

Third

```
# ReadInput: 1, Cypher: 0100001100100001
# Read input from: 0100
# Result:
#                          00000111
#
step -out -current
# Break in Module testbench_cypher_detect
step -out -current
# ReadInput: 0, Cypher: 0100001100100001
# Read input from: 0100
# Result:
#                          00000111
#
```

As it seen, it skipped one of the inputs due to clock sync problem.