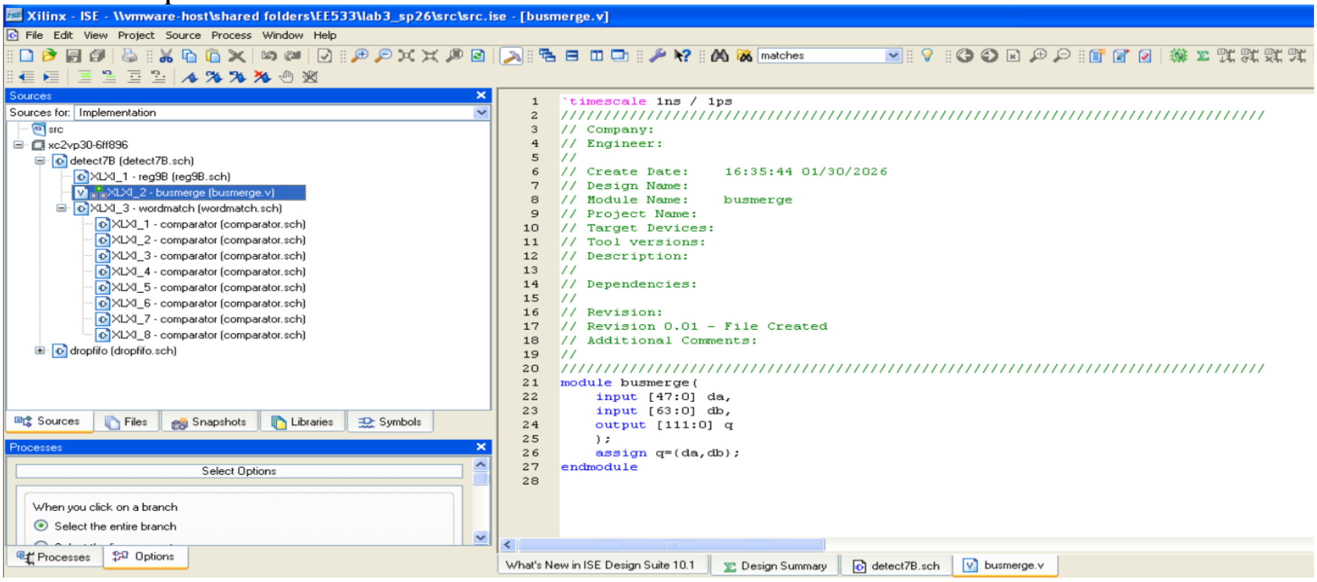
EE 533- LAB 3

Name: Chaitanya Joshi
USC ID: 4991436817 (joshic@usc.edu)
GitHub repo: https://github.com/chaijosh/sp2026_ee533_lab3

PART 1: Schematic & Verilog design for mini-IDE

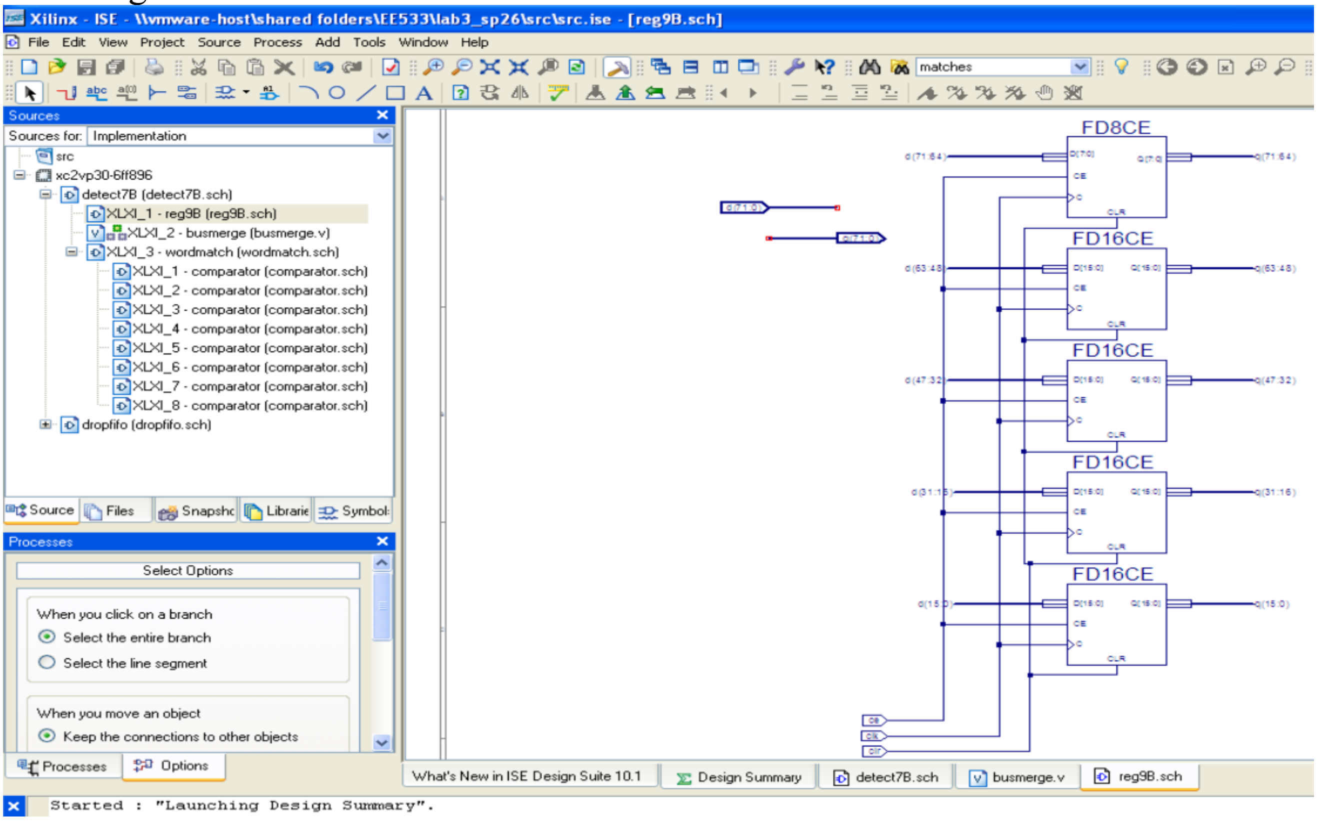
1. All schematics files are present in /main/src.

- comparator.sch


Xilinx - ISE - \\vmware-host\shared folders\EE533\lab3_sp26\src\src.ise - [busmerge.v]
File Edit View Project Source Process Window Help
Sources
Sources for: Implementation
src
xc2vp30-6ff896
detect7B (detect7B.sch)
XLXI_1 - reg9B (reg9B.sch)
XLXI_2 - busmerge (busmerge.v)
XLXI_3 - wordmatch (wordmatch.sch)
XLXI_1 - comparator (comparator.sch)
XLXI_2 - comparator (comparator.sch)
XLXI_3 - comparator (comparator.sch)
XLXI_4 - comparator (comparator.sch)
XLXI_5 - comparator (comparator.sch)
XLXI_6 - comparator (comparator.sch)
XLXI_7 - comparator (comparator.sch)
XLXI_8 - comparator (comparator.sch)
dropfifo (dropfifo.sch)
Sources
Files
Snapshots
Libraries
Symbols
Processes
Select Options
When you click on a branch
Select the entire branch
Processes
Options
What's New in ISE Design Suite 10.1
Design Summary
detect7B.sch
busmerge.v

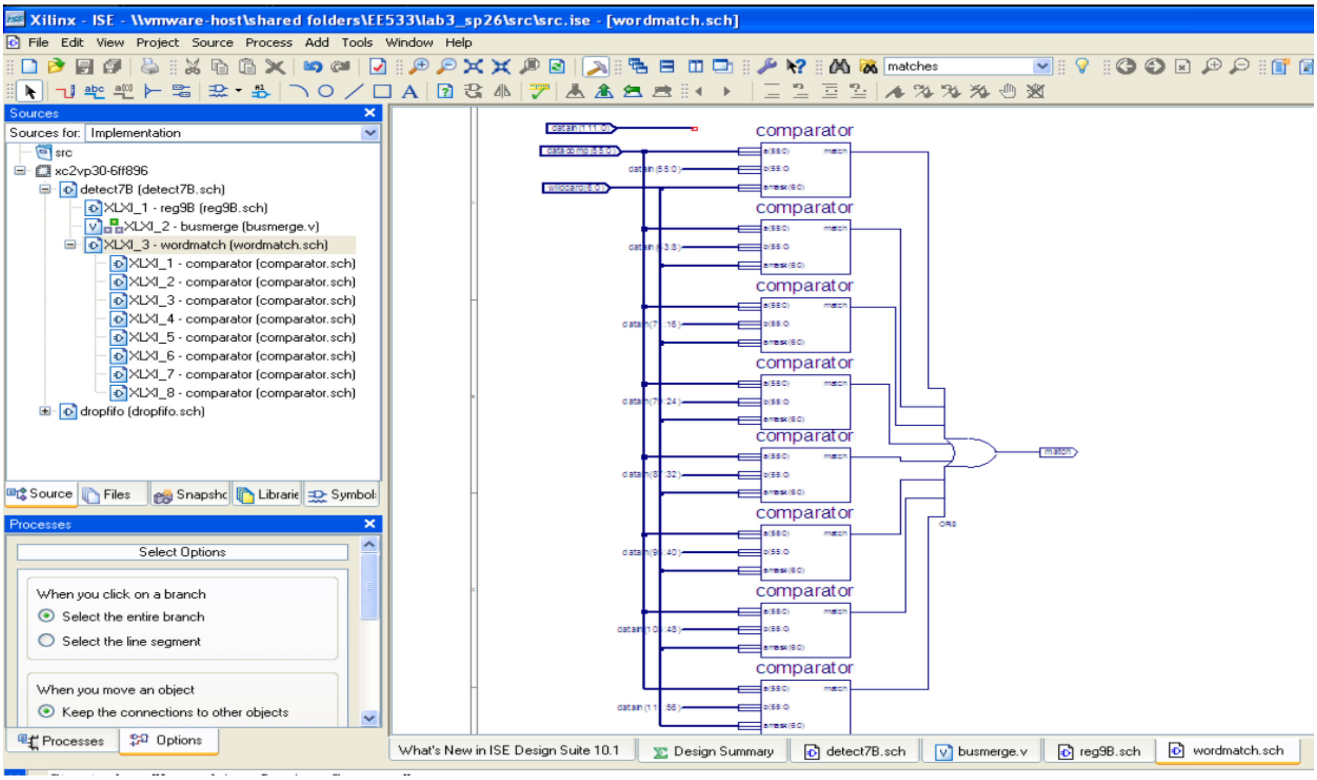
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    16:35:44 01/30/2026
// Design Name:
// Module Name:    busmerge
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module busmerge(
    input [47:0] da,
    input [63:0] db,
    output [111:0] q
    );
    assign q={da,db};
endmodule
```

- reg9B.sch


Xilinx - ISE - \\vmware-host\shared folders\EE533\lab3_sp26\src\src.ise - [reg9B.sch]
File Edit View Project Source Process Add Tools Window Help
Sources
Sources for: Implementation
src
xc2vp30-6ff896
detect7B (detect7B.sch)
XLXI_1 - reg9B (reg9B.sch)
XLXI_2 - busmerge (busmerge.v)
XLXI_3 - wordmatch (wordmatch.sch)
XLXI_1 - comparator (comparator.sch)
XLXI_2 - comparator (comparator.sch)
XLXI_3 - comparator (comparator.sch)
XLXI_4 - comparator (comparator.sch)
XLXI_5 - comparator (comparator.sch)
XLXI_6 - comparator (comparator.sch)
XLXI_7 - comparator (comparator.sch)
XLXI_8 - comparator (comparator.sch)
dropfifo (dropfifo.sch)
Source
Files
Snapshc
Librarie
Symbol
Processes
Select Options
When you click on a branch
Select the entire branch
Select the line segment
When you move an object
Keep the connections to other objects
Processes
Options
FD8CE
FD16CE
FD16CE
FD16CE
FD16CE
d(71:64)
q(71:64)
d(63:48)
q(63:48)
d(47:32)
q(47:32)
d(31:16)
q(31:16)
d(15:0)
q(15:0)
d(71:0)
q(71:0)
ce
clk
clr
What's New in ISE Design Suite 10.1
Design Summary
detect7B.sch
busmerge.v
reg9B.sch

```
Started : "Launching Design Summary".
```

- wordmatch.sch



Xilinx - ISE - \\vmware-host\shared folders\EE533\lab3_sp26\src\src.ise - [wordmatch.sch]

Sources for: Implementation

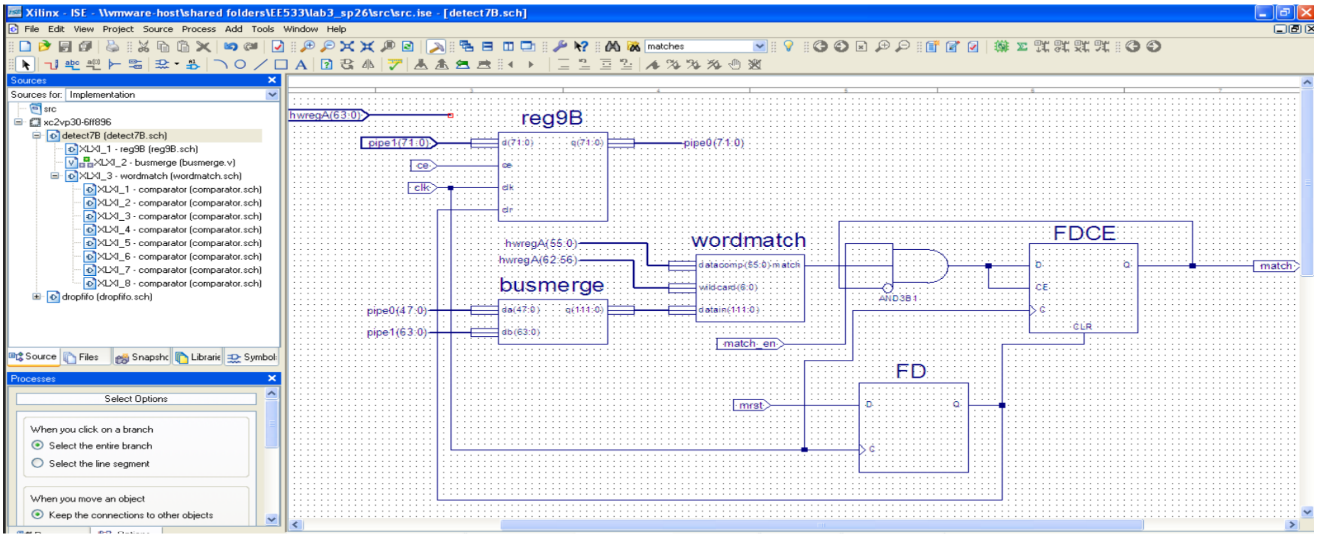
- src
- xc2vp30-6ff896
  - detect7B (detect7B.sch)
    - XLXI_1 - reg9B (reg9B.sch)
    - XLXI_2 - busmerge (busmerge.v)
    - XLXI_3 - wordmatch (wordmatch.sch)
      - XLXI_1 - comparator (comparator.sch)
      - XLXI_2 - comparator (comparator.sch)
      - XLXI_3 - comparator (comparator.sch)
      - XLXI_4 - comparator (comparator.sch)
      - XLXI_5 - comparator (comparator.sch)
      - XLXI_6 - comparator (comparator.sch)
      - XLXI_7 - comparator (comparator.sch)
      - XLXI_8 - comparator (comparator.sch)
  - dropfifo (dropfifo.sch)

Select Options

When you click on a branch
● Select the entire branch
○ Select the line segment

When you move an object
● Keep the connections to other objects

- detect7B.sch



Xilinx - ISE - \\vmware-host\shared folders\EE533\lab3_sp26\src\src.ise - [detect7B.sch]

Sources for: Implementation

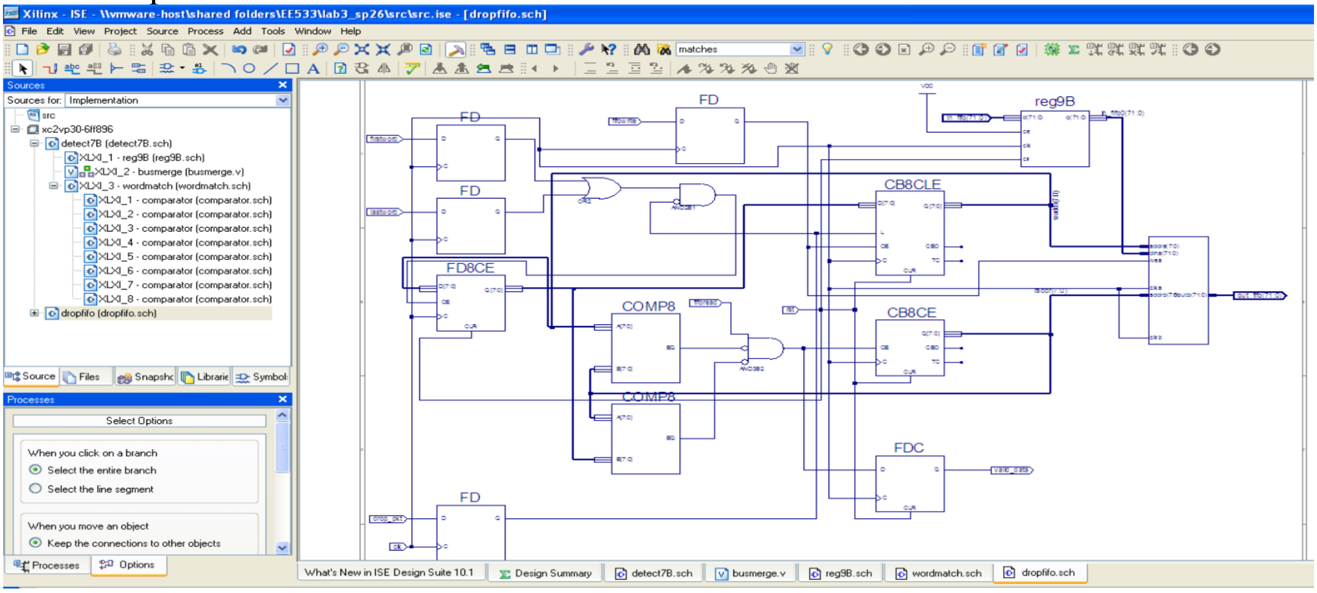
- src
- xc2vp30-6ff896
  - detect7B (detect7B.sch)
    - XLXI_1 - reg9B (reg9B.sch)
    - XLXI_2 - busmerge (busmerge.v)
    - XLXI_3 - wordmatch (wordmatch.sch)
      - XLXI_1 - comparator (comparator.sch)
      - XLXI_2 - comparator (comparator.sch)
      - XLXI_3 - comparator (comparator.sch)
      - XLXI_4 - comparator (comparator.sch)
      - XLXI_5 - comparator (comparator.sch)
      - XLXI_6 - comparator (comparator.sch)
      - XLXI_7 - comparator (comparator.sch)
      - XLXI_8 - comparator (comparator.sch)
  - dropfifo (dropfifo.sch)

Select Options

When you click on a branch
● Select the entire branch
○ Select the line segment

When you move an object
● Keep the connections to other objects

- dropfifo.sch



Xilinx - ISE - \\vmware-host\shared folders\EE533\lab3_sp26\src\src.ise - [dropfifo.sch]

Sources for: Implementation

```
src
xc2vp30-6ff896
  detect7B (detect7B.sch)
    XLXI_1 - reg9B (reg9B.sch)
    XLXI_2 - busmerge (busmerge.v)
    XLXI_3 - wordmatch (wordmatch.sch)
      XLXI_1 - comparator (comparator.sch)
      XLXI_2 - comparator (comparator.sch)
      XLXI_3 - comparator (comparator.sch)
      XLXI_4 - comparator (comparator.sch)
      XLXI_5 - comparator (comparator.sch)
      XLXI_6 - comparator (comparator.sch)
      XLXI_7 - comparator (comparator.sch)
      XLXI_8 - comparator (comparator.sch)
  dropfifo (dropfifo.sch)
```

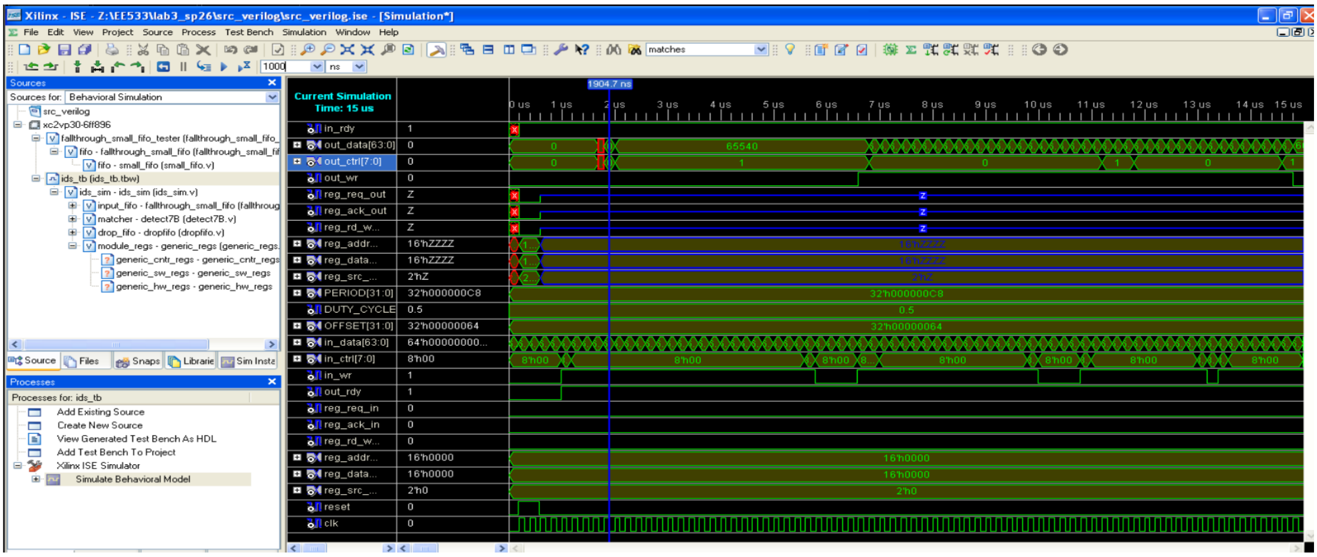
Select Options

When you click on a branch
● Select the entire branch
○ Select the line segment

When you move an object
● Keep the connections to other objects

2. Converted Verilog codes (/main/src_verilog) from schematics: APPENDIX-A

- Testbench simulation results:



Xilinx - ISE - Z:\EE533\lab3_sp26\src_verilog\src_verilog.ise - [Simulation*]

Sources for: Behavioral Simulation

```
src_verilog
xc2vp30-6ff896
  fallthrough_small_fifo_tester (fallthrough_small_fifo_
    fifo - fallthrough_small_fifo (fallthrough_small_fif
      fifo - small_fifo (small_fifo.v)
  ids_tb (ids_tb.tbw)
    ids_sim - ids_sim (ids_sim.v)
      input_fifo - fallthrough_small_fifo (fallthroug
      matcher - detect7B (detect7B.v)
      drop_fifo - dropfifo (dropfifo.v)
      module_regs - generic_regs (generic_regs.
        generic_cntr_regs - generic_cntr_regs
        generic_sw_regs - generic_sw_regs
        generic_hw_regs - generic_hw_regs
```

Processes for: ids_tb

```
Add Existing Source
Create New Source
View Generated Test Bench As HDL
Add Test Bench To Project
Xilinx ISE Simulator
  Simulate Behavioral Model
```

Current Simulation Time: 15 us

```
in_rdy            1
out_data[63:0]    0
out_ctrl[7:0]     0
out_wr            0
reg_req_out       Z
reg_ack_out       Z
reg_rd_w...       Z
reg_addr...       16'hZZZZ
reg_data...       16'hZZZZ
reg_src_...       2'hZ
PERIOD[31:0]      32'h000000C8
DUTY_CYCLE        0.5
OFFSET[31:0]      32'h00000064
in_data[63:0]     64'h00000000...
in_ctrl[7:0]      8'h00
in_wr             1
out_rdy           1
reg_req_in        0
reg_ack_in        0
reg_rd_w...       0
reg_addr...       16'h0000
reg_data...       16'h0000
reg_src_...       2'h0
reset             0
clk               0
```

PART 2: Questions

1. Explain the pattern matching algorithm in the report

- The algorithm uses sliding window method of matching a 8 byte input (56 bit data + 8bit control signal) over a window of 112 bit wide sampled data.
- For each 56 bit portion of the sampled data, the comparators compare each byte with with corresponding input byte and assert a “match” signal if the pattern is detected.

2. Take a look at the created Verilog. Do they make sense? Which do you think easier: entering the schematics or writing Verilog? Why? In which cases might you do the other?

- Many schematics have used in built modules, which are considered as blackboxes unless you view the schematics for them.
- It is difficult to keep track of wires and module instances and their random generated names, as schematics use structural programming methodology to generate HDL code. Hence, readability suffers in the converted HDL codes.
- One may write behavioural programmes for these modules which are C-like codes and a bit easier to read, however at the expense of potentially unoptimized FPGA implementation.
- Conclusion- Schematics can be used to design basic modules, and Verilog structural/behavioural codes can be written as a top module to combine different HW blocks

3. Extract ids_sim directory to your ISE project. You should now be able to simulate the mini-IDS using the ids_tb.tbw testbench. The other files are needed to emulate the pieces of the NetFPGA that are around your design. Run the testbench and take a screen shot. Describe what the testbench does and how it shows that the mini-IDS is functioning.

- The testbench emulates the NetFPGA environment with an 8 byte (+ 1byte control signal) input data and tries to match a 64 bit pattern to the input stream.
- Inputs are fed using an input FIFO. When appropriate control signals are asserted, the output of FIFP flows into the pattern matcher (detect7B).
- If pattern is detected, match=1 and the 8B input is dropped. For this, the dropfifo receives a pkt_drop signal, and if match is detected, the input is not written into the memory.

4. What is the purpose of AMASK[6:0]?

- AMASK[6:0] is used for masking (i.e. ignoring) a particular byte filed. For a 9 Byte input to the comparator, this signal acts as an enable signal for each byte, indicating whether we need to consider comparison of the corresponding input byte or not.

5. What exactly does busmerge.v do?

- It merges 2 input data of 64 bit wide (but ignores 8 bits, or "headers").
- This is required to feed into the detect7B module as a feature to detect a pattern even if it is split between 2 continuous inputs.

6. What do the comp8 modules do in this schematic?

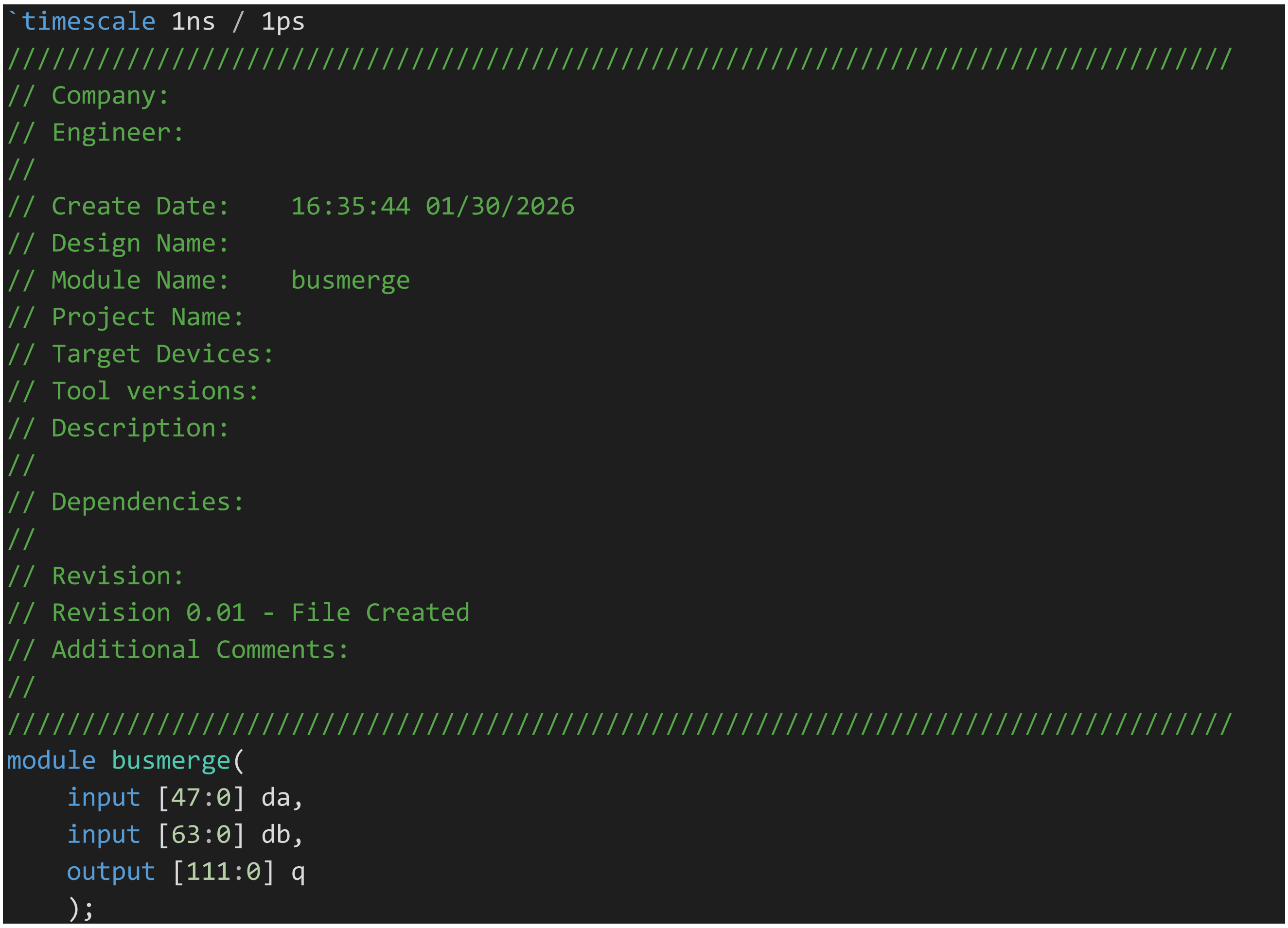
- Perform byte-wise comparison of input data with a predefined pattern.

7. What is the purpose of dual9Bmem in dropfifo.sch?

- Used as a FIFO buffer for forwarding input data after pattern matching
- The dual port feature allows simultaneous read and write operation (as per features of a FIFO).

APPENDIX-A: Verilog code dump

1. busmerge.v

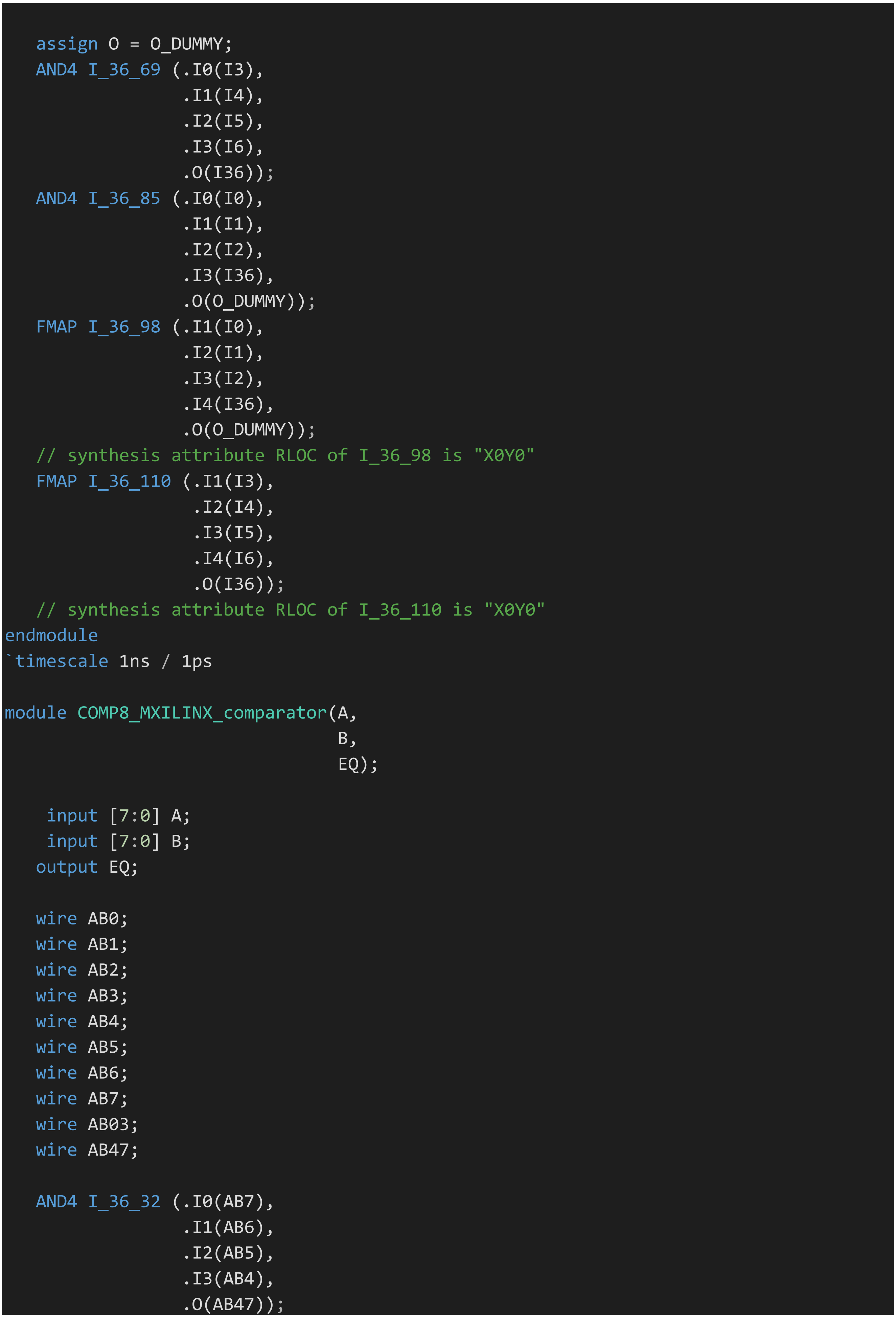


```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    16:35:44 01/30/2026
// Design Name:
// Module Name:    busmerge
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module busmerge(
    input [47:0] da,
    input [63:0] db,
    output [111:0] q
    );
```

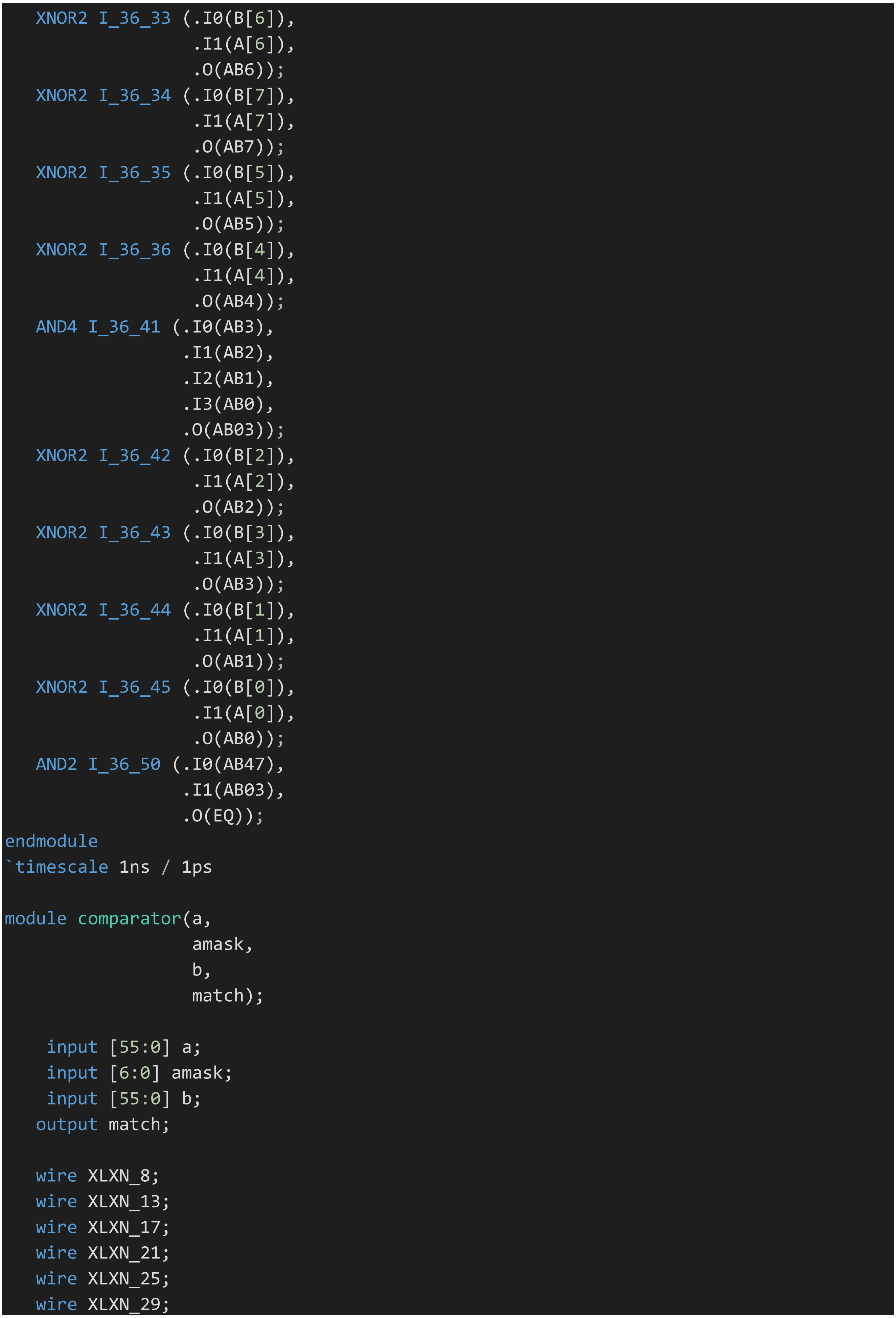
```
     assign q={da,db};
endmodule
```

2. comparator.v

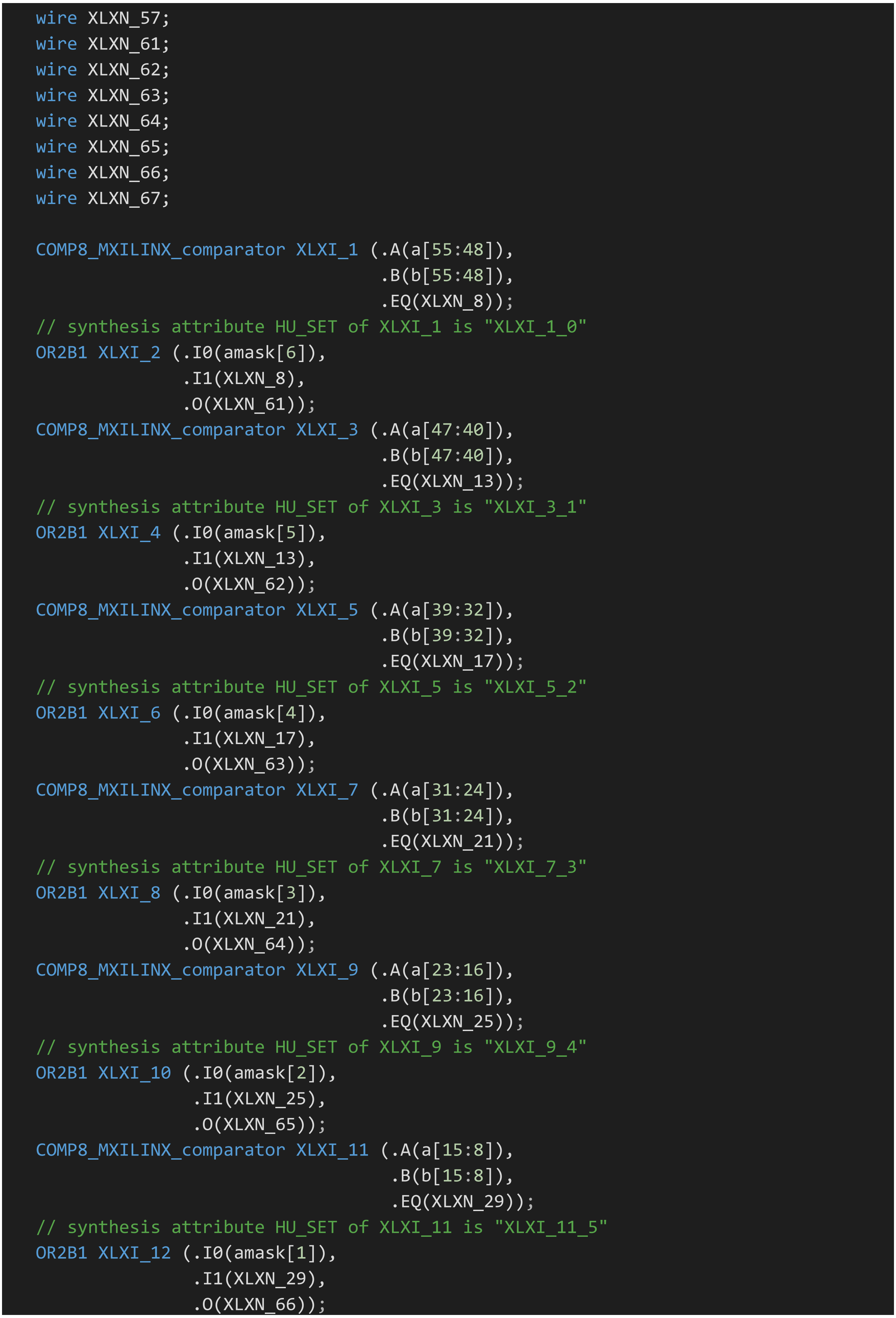
```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : comparator.vf
// /___/   /\     Timestamp : 01/30/2026 20:43:34
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w Z:/EE533/lab3_sp26/src/comparator.sch comparator.vf
//Design Name: comparator
//Device: virtex2p
//Purpose:
//    This verilog netlist is translated from an ECS schematic.It can be
//    synthesized and simulated, but it should not be modified.
//
`timescale 1ns / 1ps

module AND7_MXILINX_comparator(I0,
                               I1,
                               I2,
                               I3,
                               I4,
                               I5,
                               I6,
                               O);

    input I0;
    input I1;
    input I2;
    input I3;
    input I4;
    input I5;
    input I6;
   output O;

   wire I36;
   wire O_DUMMY;
```



```
   assign O = O_DUMMY;
   AND4 I_36_69 (.I0(I3),
                 .I1(I4),
                 .I2(I5),
                 .I3(I6),
                 .O(I36));
   AND4 I_36_85 (.I0(I0),
                 .I1(I1),
                 .I2(I2),
                 .I3(I36),
                 .O(O_DUMMY));
   FMAP I_36_98 (.I1(I0),
                 .I2(I1),
                 .I3(I2),
                 .I4(I36),
                 .O(O_DUMMY));
   // synthesis attribute RLOC of I_36_98 is "X0Y0"
   FMAP I_36_110 (.I1(I3),
                  .I2(I4),
                  .I3(I5),
                  .I4(I6),
                  .O(I36));
   // synthesis attribute RLOC of I_36_110 is "X0Y0"
endmodule
`timescale 1ns / 1ps

module COMP8_MXILINX_comparator(A,
                                B,
                                EQ);

    input [7:0] A;
    input [7:0] B;
   output EQ;

   wire AB0;
   wire AB1;
   wire AB2;
   wire AB3;
   wire AB4;
   wire AB5;
   wire AB6;
   wire AB7;
   wire AB03;
   wire AB47;

   AND4 I_36_32 (.I0(AB7),
                 .I1(AB6),
                 .I2(AB5),
                 .I3(AB4),
                 .O(AB47));
```



```
   XNOR2 I_36_33 (.I0(B[6]), 
                  .I1(A[6]), 
                  .O(AB6));
   XNOR2 I_36_34 (.I0(B[7]), 
                  .I1(A[7]), 
                  .O(AB7));
   XNOR2 I_36_35 (.I0(B[5]), 
                  .I1(A[5]), 
                  .O(AB5));
   XNOR2 I_36_36 (.I0(B[4]), 
                  .I1(A[4]), 
                  .O(AB4));
   AND4 I_36_41 (.I0(AB3), 
                 .I1(AB2), 
                 .I2(AB1), 
                 .I3(AB0), 
                 .O(AB03));
   XNOR2 I_36_42 (.I0(B[2]), 
                  .I1(A[2]), 
                  .O(AB2));
   XNOR2 I_36_43 (.I0(B[3]), 
                  .I1(A[3]), 
                  .O(AB3));
   XNOR2 I_36_44 (.I0(B[1]), 
                  .I1(A[1]), 
                  .O(AB1));
   XNOR2 I_36_45 (.I0(B[0]), 
                  .I1(A[0]), 
                  .O(AB0));
   AND2 I_36_50 (.I0(AB47), 
                 .I1(AB03), 
                 .O(EQ));
endmodule
`timescale 1ns / 1ps

module comparator(a, 
                  amask, 
                  b, 
                  match);

    input [55:0] a;
    input [6:0] amask;
    input [55:0] b;
   output match;

   wire XLXN_8;
   wire XLXN_13;
   wire XLXN_17;
   wire XLXN_21;
   wire XLXN_25;
   wire XLXN_29;
```



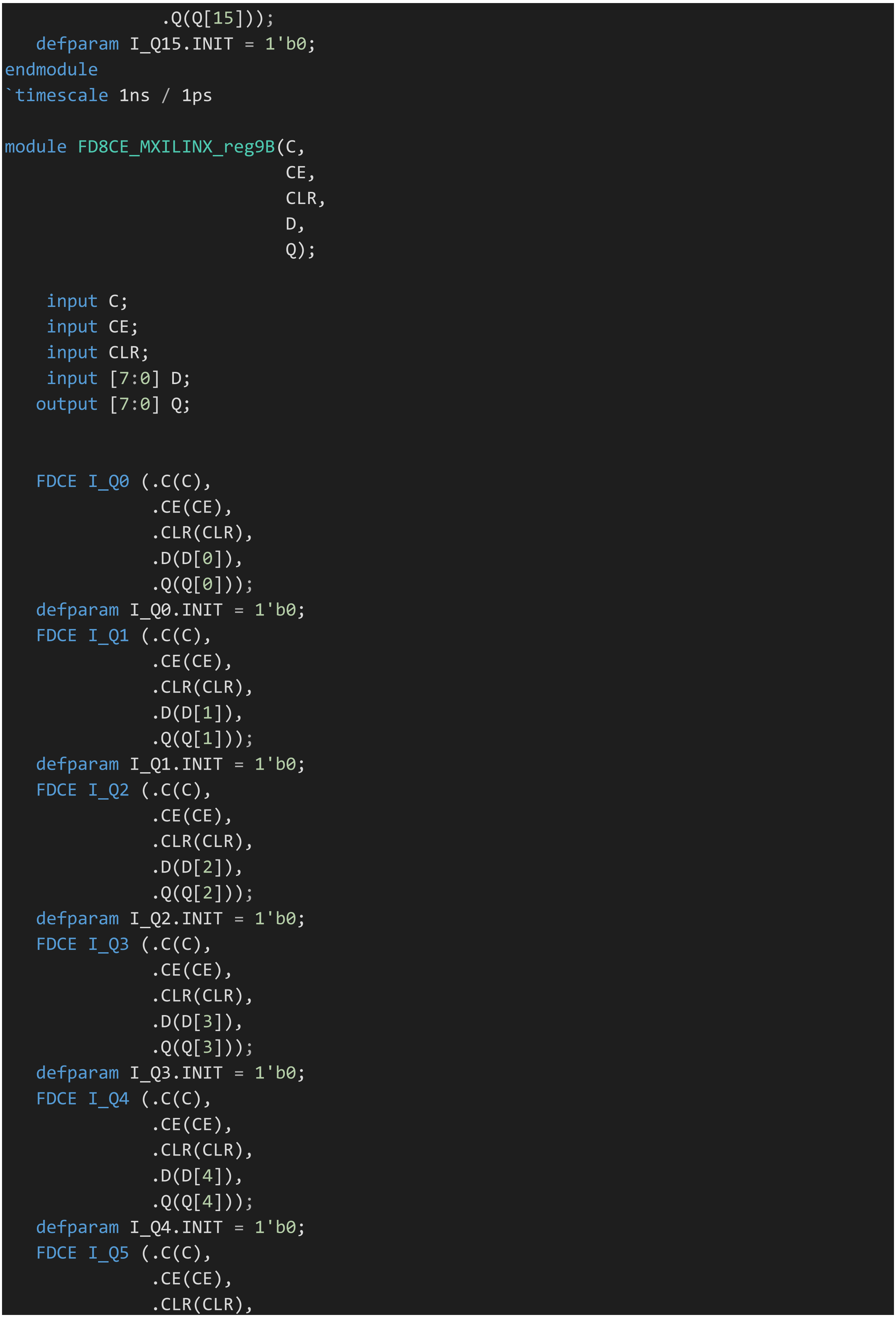
```
wire XLXN_57;
wire XLXN_61;
wire XLXN_62;
wire XLXN_63;
wire XLXN_64;
wire XLXN_65;
wire XLXN_66;
wire XLXN_67;

COMP8_MXILINX_comparator XLXI_1 (.A(a[55:48]),
                                 .B(b[55:48]),
                                 .EQ(XLXN_8));
// synthesis attribute HU_SET of XLXI_1 is "XLXI_1_0"
OR2B1 XLXI_2 (.I0(amask[6]),
              .I1(XLXN_8),
              .O(XLXN_61));
COMP8_MXILINX_comparator XLXI_3 (.A(a[47:40]),
                                 .B(b[47:40]),
                                 .EQ(XLXN_13));
// synthesis attribute HU_SET of XLXI_3 is "XLXI_3_1"
OR2B1 XLXI_4 (.I0(amask[5]),
              .I1(XLXN_13),
              .O(XLXN_62));
COMP8_MXILINX_comparator XLXI_5 (.A(a[39:32]),
                                 .B(b[39:32]),
                                 .EQ(XLXN_17));
// synthesis attribute HU_SET of XLXI_5 is "XLXI_5_2"
OR2B1 XLXI_6 (.I0(amask[4]),
              .I1(XLXN_17),
              .O(XLXN_63));
COMP8_MXILINX_comparator XLXI_7 (.A(a[31:24]),
                                 .B(b[31:24]),
                                 .EQ(XLXN_21));
// synthesis attribute HU_SET of XLXI_7 is "XLXI_7_3"
OR2B1 XLXI_8 (.I0(amask[3]),
              .I1(XLXN_21),
              .O(XLXN_64));
COMP8_MXILINX_comparator XLXI_9 (.A(a[23:16]),
                                 .B(b[23:16]),
                                 .EQ(XLXN_25));
// synthesis attribute HU_SET of XLXI_9 is "XLXI_9_4"
OR2B1 XLXI_10 (.I0(amask[2]),
               .I1(XLXN_25),
               .O(XLXN_65));
COMP8_MXILINX_comparator XLXI_11 (.A(a[15:8]),
                                  .B(b[15:8]),
                                  .EQ(XLXN_29));
// synthesis attribute HU_SET of XLXI_11 is "XLXI_11_5"
OR2B1 XLXI_12 (.I0(amask[1]),
               .I1(XLXN_29),
               .O(XLXN_66));
```

```
   COMP8_MXILINX_comparator XLXI_25 (.A(a[7:0]), 
                                     .B(b[7:0]), 
                                     .EQ(XLXN_57));
   // synthesis attribute HU_SET of XLXI_25 is "XLXI_25_6"
   OR2B1 XLXI_26 (.I0(amask[0]), 
                  .I1(XLXN_57), 
                  .O(XLXN_67));
   AND7_MXILINX_comparator XLXI_27 (.I0(XLXN_67), 
                                    .I1(XLXN_66), 
                                    .I2(XLXN_65), 
                                    .I3(XLXN_64), 
                                    .I4(XLXN_63), 
                                    .I5(XLXN_62), 
                                    .I6(XLXN_61), 
                                    .O(match));
   // synthesis attribute HU_SET of XLXI_27 is "XLXI_27_7"
endmodule
```

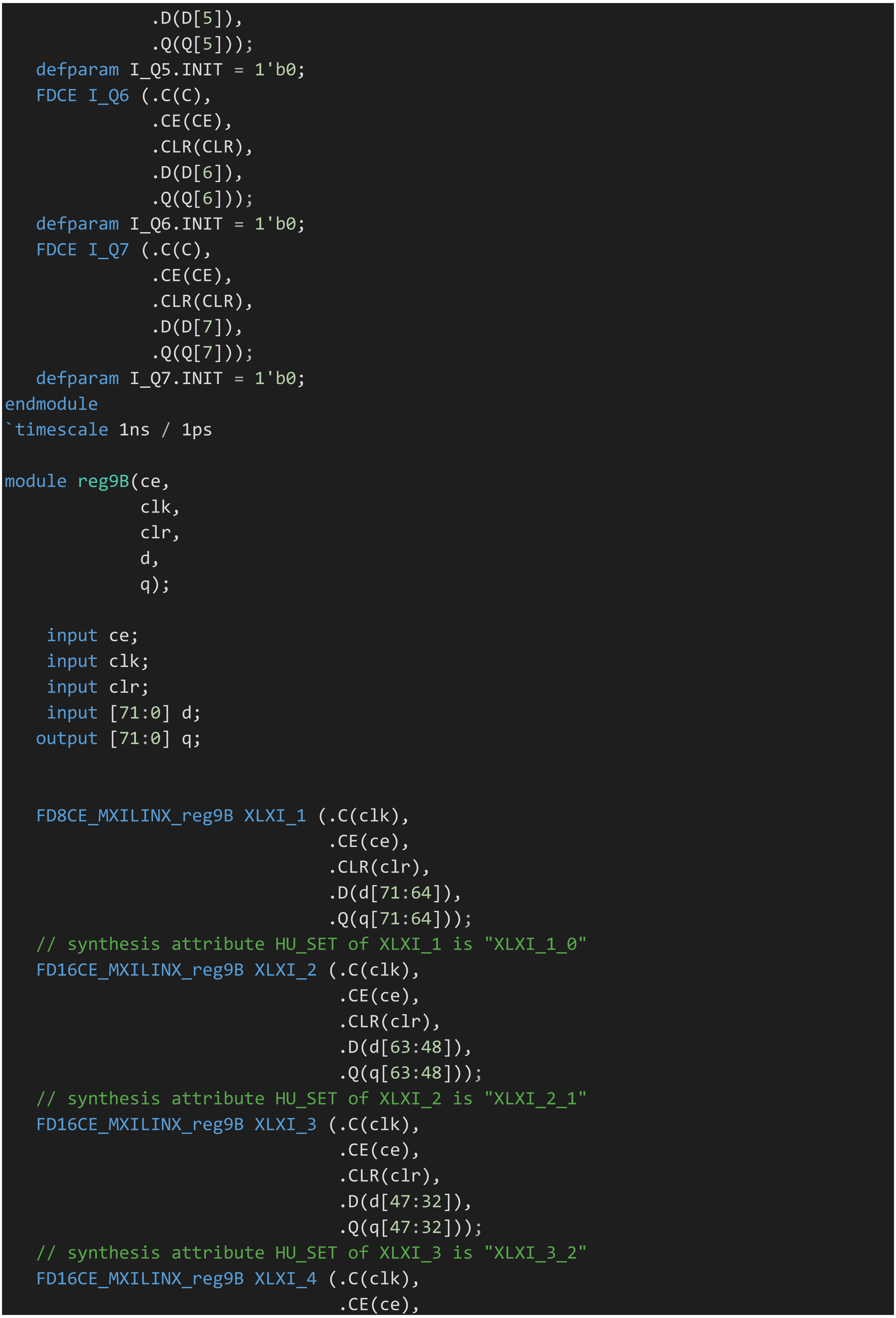
3. reg9B.v

```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____ 
//  /   /\/   / 
// /___/  \  /    Vendor: Xilinx 
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : reg9B.vf
// /___/   /\     Timestamp : 01/30/2026 20:43:34
// \   \  /  \ 
//  \___\/\___\ 
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w Z:/EE533/lab3_sp26/src/reg9B.sch reg9B.vf
//Design Name: reg9B
//Device: virtex2p
//Purpose:
//    This verilog netlist is translated from an ECS schematic.It can be 
//    synthesized and simulated, but it should not be modified. 
//
`timescale 1ns / 1ps

module FD16CE_MXILINX_reg9B(C, 
                            CE, 
                            CLR, 
                            D, 
                            Q);
```

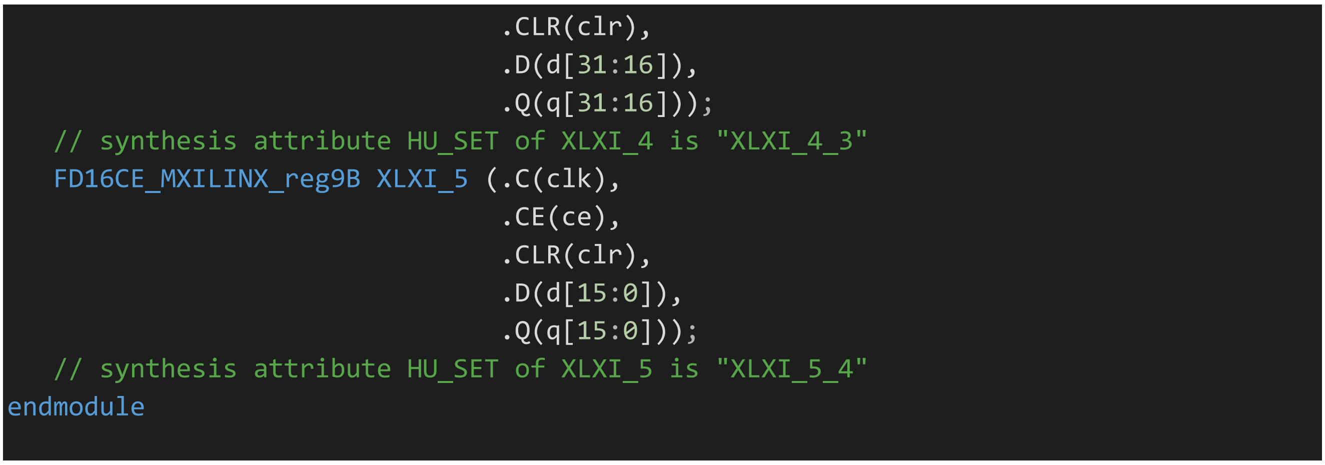
```
 input C;
 input CE;
 input CLR;
 input [15:0] D;
output [15:0] Q;


FDCE I_Q0 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[0]),
           .Q(Q[0]));
defparam I_Q0.INIT = 1'b0;
FDCE I_Q1 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[1]),
           .Q(Q[1]));
defparam I_Q1.INIT = 1'b0;
FDCE I_Q2 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[2]),
           .Q(Q[2]));
defparam I_Q2.INIT = 1'b0;
FDCE I_Q3 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[3]),
           .Q(Q[3]));
defparam I_Q3.INIT = 1'b0;
FDCE I_Q4 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[4]),
           .Q(Q[4]));
defparam I_Q4.INIT = 1'b0;
FDCE I_Q5 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[5]),
           .Q(Q[5]));
defparam I_Q5.INIT = 1'b0;
FDCE I_Q6 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[6]),
           .Q(Q[6]));
defparam I_Q6.INIT = 1'b0;
FDCE I_Q7 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[7]),
           .Q(Q[7]));
defparam I_Q7.INIT = 1'b0;
FDCE I_Q8 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[8]),
           .Q(Q[8]));
defparam I_Q8.INIT = 1'b0;
FDCE I_Q9 (.C(C),
           .CE(CE),
           .CLR(CLR),
           .D(D[9]),
           .Q(Q[9]));
defparam I_Q9.INIT = 1'b0;
FDCE I_Q10 (.C(C),
            .CE(CE),
            .CLR(CLR),
            .D(D[10]),
            .Q(Q[10]));
defparam I_Q10.INIT = 1'b0;
FDCE I_Q11 (.C(C),
            .CE(CE),
            .CLR(CLR),
            .D(D[11]),
            .Q(Q[11]));
defparam I_Q11.INIT = 1'b0;
FDCE I_Q12 (.C(C),
            .CE(CE),
            .CLR(CLR),
            .D(D[12]),
            .Q(Q[12]));
defparam I_Q12.INIT = 1'b0;
FDCE I_Q13 (.C(C),
            .CE(CE),
            .CLR(CLR),
            .D(D[13]),
            .Q(Q[13]));
defparam I_Q13.INIT = 1'b0;
FDCE I_Q14 (.C(C),
            .CE(CE),
            .CLR(CLR),
            .D(D[14]),
            .Q(Q[14]));
defparam I_Q14.INIT = 1'b0;
FDCE I_Q15 (.C(C),
            .CE(CE),
            .CLR(CLR),
            .D(D[15]),
```



```
               .Q(Q[15]));
   defparam I_Q15.INIT = 1'b0;
endmodule
`timescale 1ns / 1ps

module FD8CE_MXILINX_reg9B(C,
                           CE,
                           CLR,
                           D,
                           Q);

    input C;
    input CE;
    input CLR;
    input [7:0] D;
   output [7:0] Q;


   FDCE I_Q0 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[0]),
              .Q(Q[0]));
   defparam I_Q0.INIT = 1'b0;
   FDCE I_Q1 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[1]),
              .Q(Q[1]));
   defparam I_Q1.INIT = 1'b0;
   FDCE I_Q2 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[2]),
              .Q(Q[2]));
   defparam I_Q2.INIT = 1'b0;
   FDCE I_Q3 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[3]),
              .Q(Q[3]));
   defparam I_Q3.INIT = 1'b0;
   FDCE I_Q4 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[4]),
              .Q(Q[4]));
   defparam I_Q4.INIT = 1'b0;
   FDCE I_Q5 (.C(C),
              .CE(CE),
              .CLR(CLR),
```

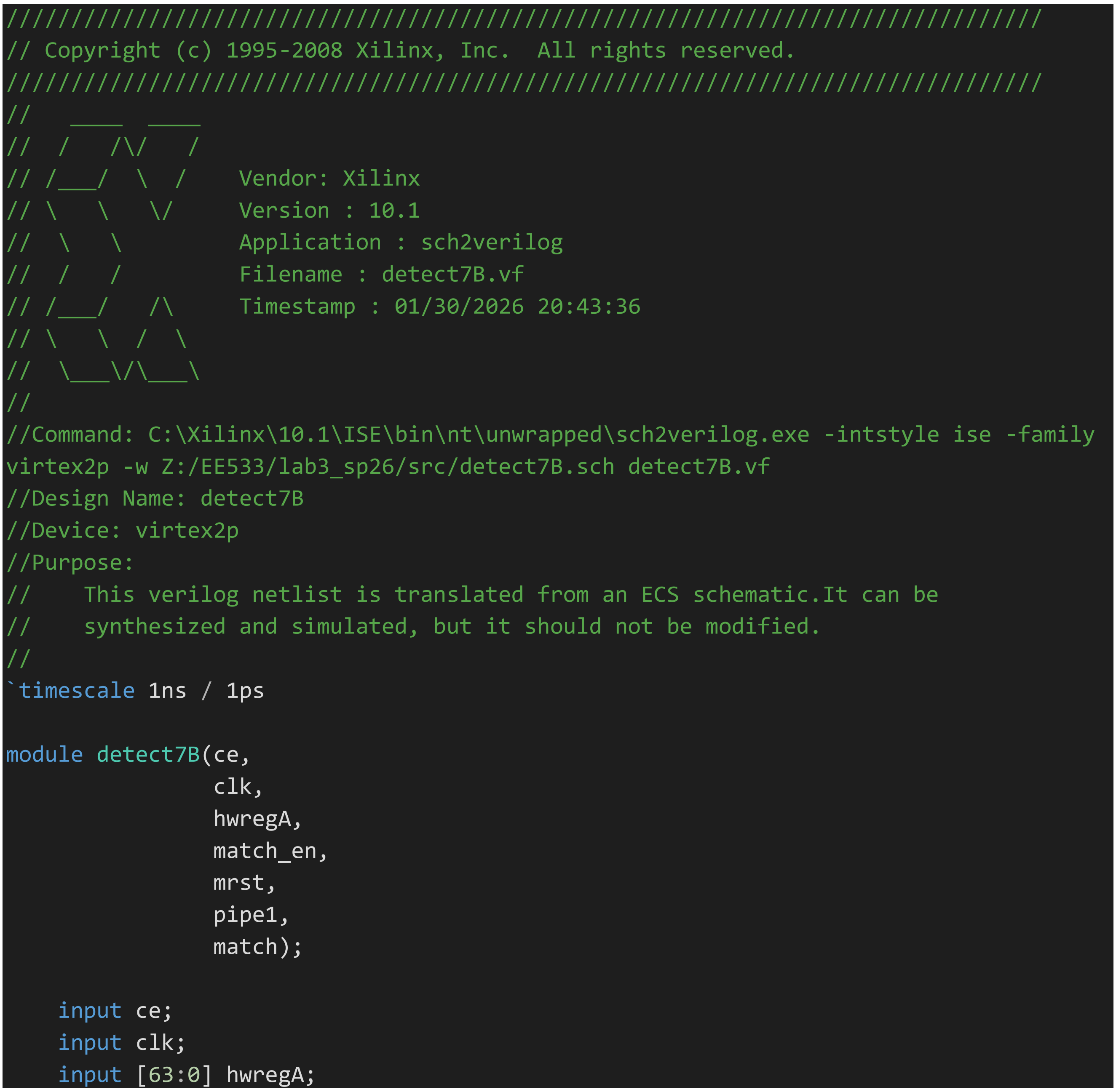


```
              .D(D[5]),
              .Q(Q[5]));
   defparam I_Q5.INIT = 1'b0;
   FDCE I_Q6 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[6]),
              .Q(Q[6]));
   defparam I_Q6.INIT = 1'b0;
   FDCE I_Q7 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[7]),
              .Q(Q[7]));
   defparam I_Q7.INIT = 1'b0;
endmodule
`timescale 1ns / 1ps

module reg9B(ce,
             clk,
             clr,
             d,
             q);

    input ce;
    input clk;
    input clr;
    input [71:0] d;
   output [71:0] q;


   FD8CE_MXILINX_reg9B XLXI_1 (.C(clk),
                               .CE(ce),
                               .CLR(clr),
                               .D(d[71:64]),
                               .Q(q[71:64]));
   // synthesis attribute HU_SET of XLXI_1 is "XLXI_1_0"
   FD16CE_MXILINX_reg9B XLXI_2 (.C(clk),
                                .CE(ce),
                                .CLR(clr),
                                .D(d[63:48]),
                                .Q(q[63:48]));
   // synthesis attribute HU_SET of XLXI_2 is "XLXI_2_1"
   FD16CE_MXILINX_reg9B XLXI_3 (.C(clk),
                                .CE(ce),
                                .CLR(clr),
                                .D(d[47:32]),
                                .Q(q[47:32]));
   // synthesis attribute HU_SET of XLXI_3 is "XLXI_3_2"
   FD16CE_MXILINX_reg9B XLXI_4 (.C(clk),
                                .CE(ce),
```



```
                                .CLR(clr), 
                                .D(d[31:16]), 
                                .Q(q[31:16]));
   // synthesis attribute HU_SET of XLXI_4 is "XLXI_4_3"
   FD16CE_MXILINX_reg9B XLXI_5 (.C(clk), 
                                .CE(ce), 
                                .CLR(clr), 
                                .D(d[15:0]), 
                                .Q(q[15:0]));
   // synthesis attribute HU_SET of XLXI_5 is "XLXI_5_4"
endmodule
```

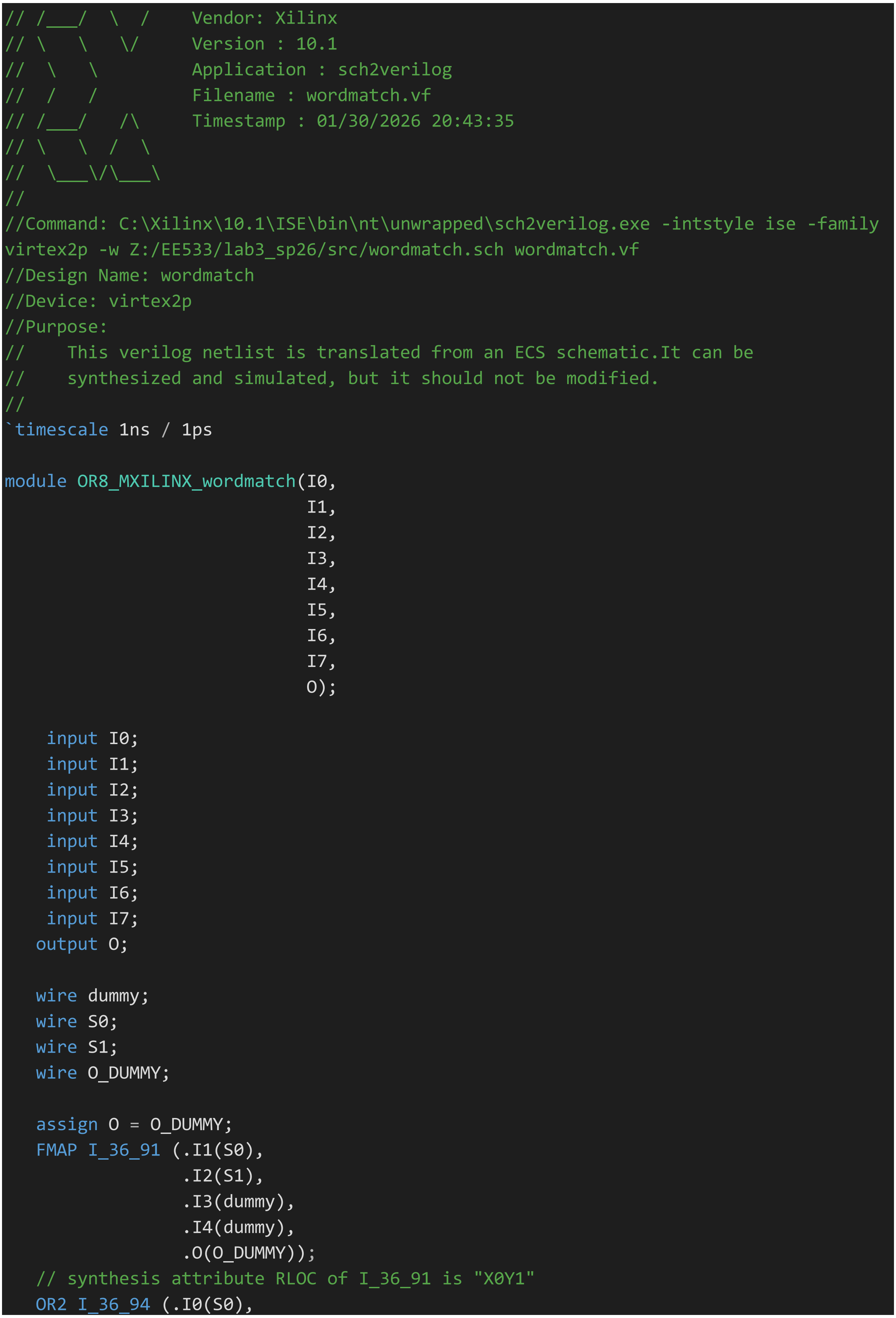
4. detect7B.v



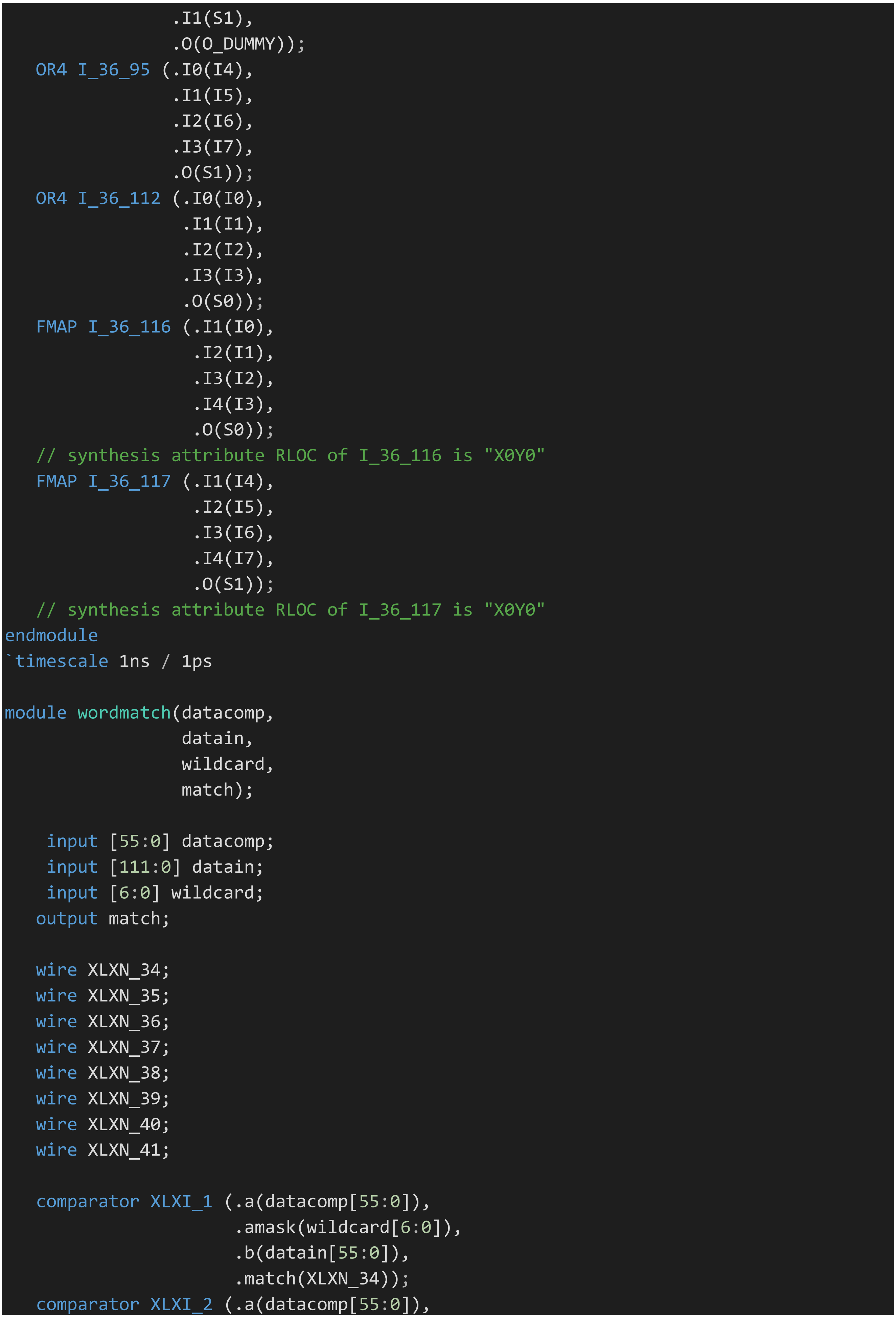
```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____ 
//  /   /\/   / 
// /___/  \  /    Vendor: Xilinx 
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : detect7B.vf
// /___/   /\     Timestamp : 01/30/2026 20:43:36
// \   \  /  \ 
//  \___\/\___\ 
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w Z:/EE533/lab3_sp26/src/detect7B.sch detect7B.vf
//Design Name: detect7B
//Device: virtex2p
//Purpose:
//    This verilog netlist is translated from an ECS schematic.It can be 
//    synthesized and simulated, but it should not be modified. 
//
`timescale 1ns / 1ps

module detect7B(ce, 
                clk, 
                hwregA, 
                match_en, 
                mrst, 
                pipe1, 
                match);

    input ce;
    input clk;
    input [63:0] hwregA;
    input match_en;
    input mrst;
    input [71:0] pipe1;
   output match;

   wire [71:0] pipe0;
   wire [111:0] XLXN_8;
   wire XLXN_9;
   wire XLXN_12;
   wire XLXN_16;
   wire match_DUMMY;

   assign match = match_DUMMY;
   reg9B XLXI_1 (.ce(ce),
                 .clk(clk),
                 .clr(XLXN_16),
                 .d(pipe1[71:0]),
                 .q(pipe0[71:0]));
   busmerge XLXI_2 (.da(pipe0[47:0]),
                    .db(pipe1[63:0]),
                    .q(XLXN_8[111:0]));
   wordmatch XLXI_3 (.datacomp(hwregA[55:0]),
                     .datain(XLXN_8[111:0]),
                     .wildcard(hwregA[62:56]),
                     .match(XLXN_9));
   FD XLXI_4 (.C(clk),
              .D(mrst),
              .Q(XLXN_16));
   defparam XLXI_4.INIT = 1'b0;
   FDCE XLXI_5 (.C(clk),
                .CE(XLXN_12),
                .CLR(XLXN_16),
                .D(XLXN_12),
                .Q(match_DUMMY));
   defparam XLXI_5.INIT = 1'b0;
   AND3B1 XLXI_6 (.I0(match_DUMMY),
                  .I1(XLXN_9),
                  .I2(match_en),
                  .O(XLXN_12));
endmodule
```

5. wordmatch.v

```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
```



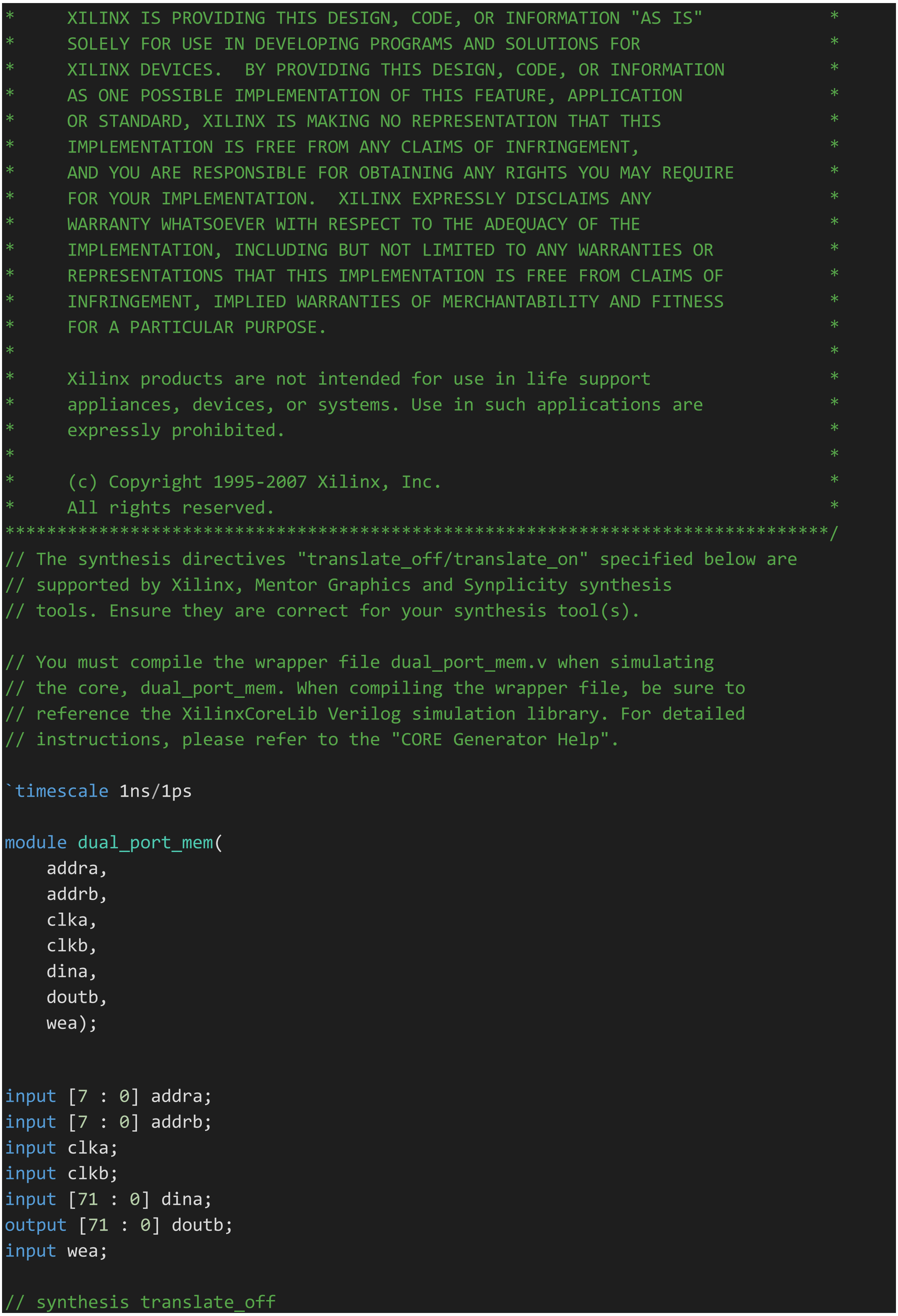
```
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : wordmatch.vf
// /___/   /\     Timestamp : 01/30/2026 20:43:35
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family
virtex2p -w Z:/EE533/lab3_sp26/src/wordmatch.sch wordmatch.vf
//Design Name: wordmatch
//Device: virtex2p
//Purpose:
//    This verilog netlist is translated from an ECS schematic.It can be
//    synthesized and simulated, but it should not be modified.
//
`timescale 1ns / 1ps

module OR8_MXILINX_wordmatch(I0,
                             I1,
                             I2,
                             I3,
                             I4,
                             I5,
                             I6,
                             I7,
                             O);

    input I0;
    input I1;
    input I2;
    input I3;
    input I4;
    input I5;
    input I6;
    input I7;
   output O;

   wire dummy;
   wire S0;
   wire S1;
   wire O_DUMMY;

   assign O = O_DUMMY;
   FMAP I_36_91 (.I1(S0),
                 .I2(S1),
                 .I3(dummy),
                 .I4(dummy),
                 .O(O_DUMMY));
   // synthesis attribute RLOC of I_36_91 is "X0Y1"
   OR2 I_36_94 (.I0(S0),
```



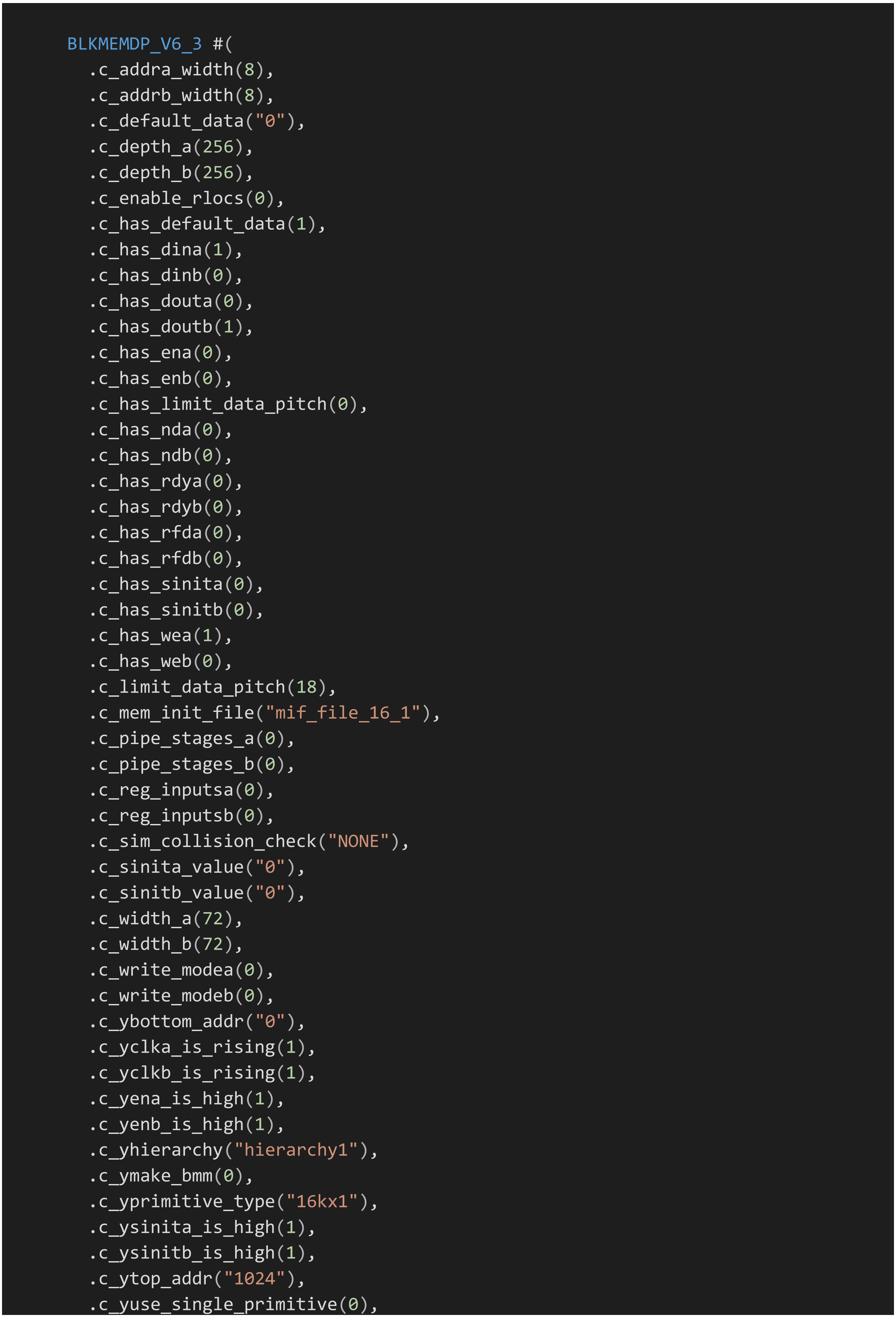
```
                .I1(S1),
                .O(O_DUMMY));
   OR4 I_36_95 (.I0(I4),
                .I1(I5),
                .I2(I6),
                .I3(I7),
                .O(S1));
   OR4 I_36_112 (.I0(I0),
                 .I1(I1),
                 .I2(I2),
                 .I3(I3),
                 .O(S0));
   FMAP I_36_116 (.I1(I0),
                  .I2(I1),
                  .I3(I2),
                  .I4(I3),
                  .O(S0));
   // synthesis attribute RLOC of I_36_116 is "X0Y0"
   FMAP I_36_117 (.I1(I4),
                  .I2(I5),
                  .I3(I6),
                  .I4(I7),
                  .O(S1));
   // synthesis attribute RLOC of I_36_117 is "X0Y0"
endmodule
`timescale 1ns / 1ps

module wordmatch(datacomp,
                 datain,
                 wildcard,
                 match);

    input [55:0] datacomp;
    input [111:0] datain;
    input [6:0] wildcard;
   output match;

   wire XLXN_34;
   wire XLXN_35;
   wire XLXN_36;
   wire XLXN_37;
   wire XLXN_38;
   wire XLXN_39;
   wire XLXN_40;
   wire XLXN_41;

   comparator XLXI_1 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[55:0]),
                      .match(XLXN_34));
   comparator XLXI_2 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[63:8]),
                      .match(XLXN_35));
   comparator XLXI_3 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[71:16]),
                      .match(XLXN_36));
   comparator XLXI_4 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[79:24]),
                      .match(XLXN_37));
   comparator XLXI_5 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[87:32]),
                      .match(XLXN_38));
   comparator XLXI_6 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[95:40]),
                      .match(XLXN_39));
   comparator XLXI_7 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[103:48]),
                      .match(XLXN_40));
   comparator XLXI_8 (.a(datacomp[55:0]),
                      .amask(wildcard[6:0]),
                      .b(datain[111:56]),
                      .match(XLXN_41));
   OR8_MXILINX_wordmatch XLXI_9 (.I0(XLXN_41),
                                 .I1(XLXN_40),
                                 .I2(XLXN_39),
                                 .I3(XLXN_38),
                                 .I4(XLXN_37),
                                 .I5(XLXN_36),
                                 .I6(XLXN_35),
                                 .I7(XLXN_34),
                                 .O(match));
   // synthesis attribute HU_SET of XLXI_9 is "XLXI_9_0"
endmodule
```

6. dual_port_mem.v (generated from IP Core)

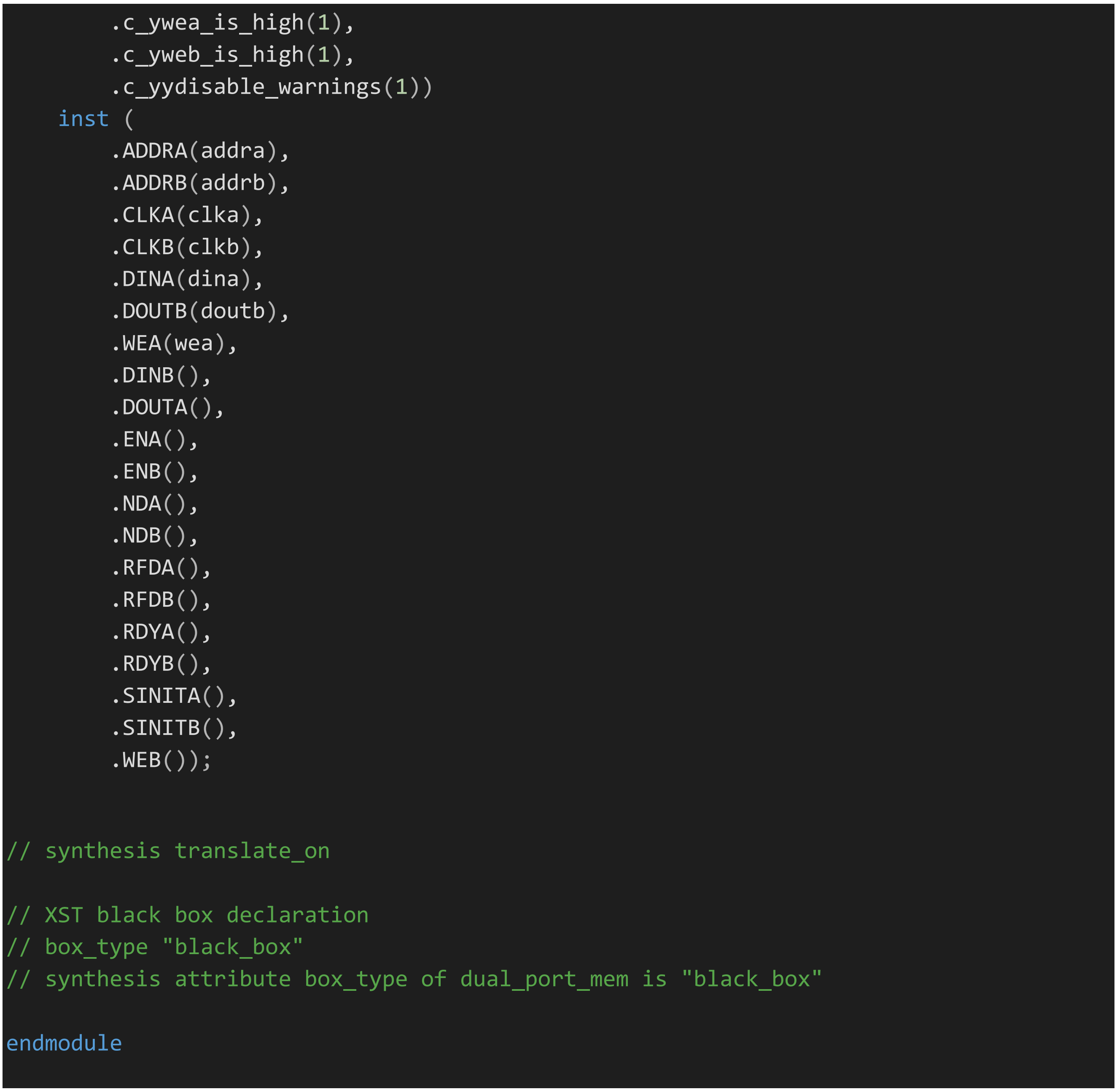
```
/*******************************************************************************
*     This file is owned and controlled by Xilinx and must be used           *
*     solely for design, simulation, implementation and creation of          *
*     design files limited to Xilinx devices or technologies. Use            *
*     with non-Xilinx devices or technologies is expressly prohibited        *
*     and immediately terminates your license.                               *
*                                                                            *
```



```
*     XILINX IS PROVIDING THIS DESIGN, CODE, OR INFORMATION "AS IS"            *
*     SOLELY FOR USE IN DEVELOPING PROGRAMS AND SOLUTIONS FOR                  *
*     XILINX DEVICES.  BY PROVIDING THIS DESIGN, CODE, OR INFORMATION          *
*     AS ONE POSSIBLE IMPLEMENTATION OF THIS FEATURE, APPLICATION              *
*     OR STANDARD, XILINX IS MAKING NO REPRESENTATION THAT THIS                *
*     IMPLEMENTATION IS FREE FROM ANY CLAIMS OF INFRINGEMENT,                  *
*     AND YOU ARE RESPONSIBLE FOR OBTAINING ANY RIGHTS YOU MAY REQUIRE         *
*     FOR YOUR IMPLEMENTATION.  XILINX EXPRESSLY DISCLAIMS ANY                 *
*     WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE                  *
*     IMPLEMENTATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR           *
*     REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF          *
*     INFRINGEMENT, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS          *
*     FOR A PARTICULAR PURPOSE.                                                *
*                                                                              *
*     Xilinx products are not intended for use in life support                 *
*     appliances, devices, or systems. Use in such applications are            *
*     expressly prohibited.                                                    *
*                                                                              *
*     (c) Copyright 1995-2007 Xilinx, Inc.                                     *
*     All rights reserved.                                                     *
*******************************************************************************/
// The synthesis directives "translate_off/translate_on" specified below are
// supported by Xilinx, Mentor Graphics and Synplicity synthesis
// tools. Ensure they are correct for your synthesis tool(s).

// You must compile the wrapper file dual_port_mem.v when simulating
// the core, dual_port_mem. When compiling the wrapper file, be sure to
// reference the XilinxCoreLib Verilog simulation library. For detailed
// instructions, please refer to the "CORE Generator Help".

`timescale 1ns/1ps

module dual_port_mem(
    addra,
    addrb,
    clka,
    clkb,
    dina,
    doutb,
    wea);


input [7 : 0] addra;
input [7 : 0] addrb;
input clka;
input clkb;
input [71 : 0] dina;
output [71 : 0] doutb;
input wea;

// synthesis translate_off
```

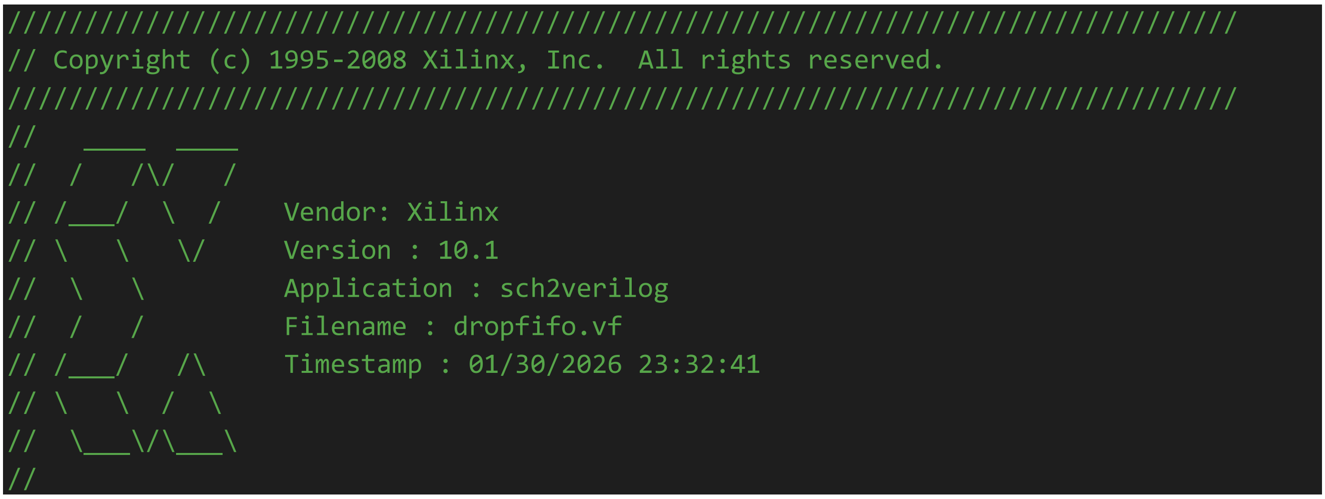


```
    BLKMEMDP_V6_3 #(
      .c_addra_width(8),
      .c_addrb_width(8),
      .c_default_data("0"),
      .c_depth_a(256),
      .c_depth_b(256),
      .c_enable_rlocs(0),
      .c_has_default_data(1),
      .c_has_dina(1),
      .c_has_dinb(0),
      .c_has_douta(0),
      .c_has_doutb(1),
      .c_has_ena(0),
      .c_has_enb(0),
      .c_has_limit_data_pitch(0),
      .c_has_nda(0),
      .c_has_ndb(0),
      .c_has_rdya(0),
      .c_has_rdyb(0),
      .c_has_rfda(0),
      .c_has_rfdb(0),
      .c_has_sinita(0),
      .c_has_sinitb(0),
      .c_has_wea(1),
      .c_has_web(0),
      .c_limit_data_pitch(18),
      .c_mem_init_file("mif_file_16_1"),
      .c_pipe_stages_a(0),
      .c_pipe_stages_b(0),
      .c_reg_inputsa(0),
      .c_reg_inputsb(0),
      .c_sim_collision_check("NONE"),
      .c_sinita_value("0"),
      .c_sinitb_value("0"),
      .c_width_a(72),
      .c_width_b(72),
      .c_write_modea(0),
      .c_write_modeb(0),
      .c_ybottom_addr("0"),
      .c_yclka_is_rising(1),
      .c_yclkb_is_rising(1),
      .c_yena_is_high(1),
      .c_yenb_is_high(1),
      .c_yhierarchy("hierarchy1"),
      .c_ymake_bmm(0),
      .c_yprimitive_type("16kx1"),
      .c_ysinita_is_high(1),
      .c_ysinitb_is_high(1),
      .c_ytop_addr("1024"),
      .c_yuse_single_primitive(0),
```

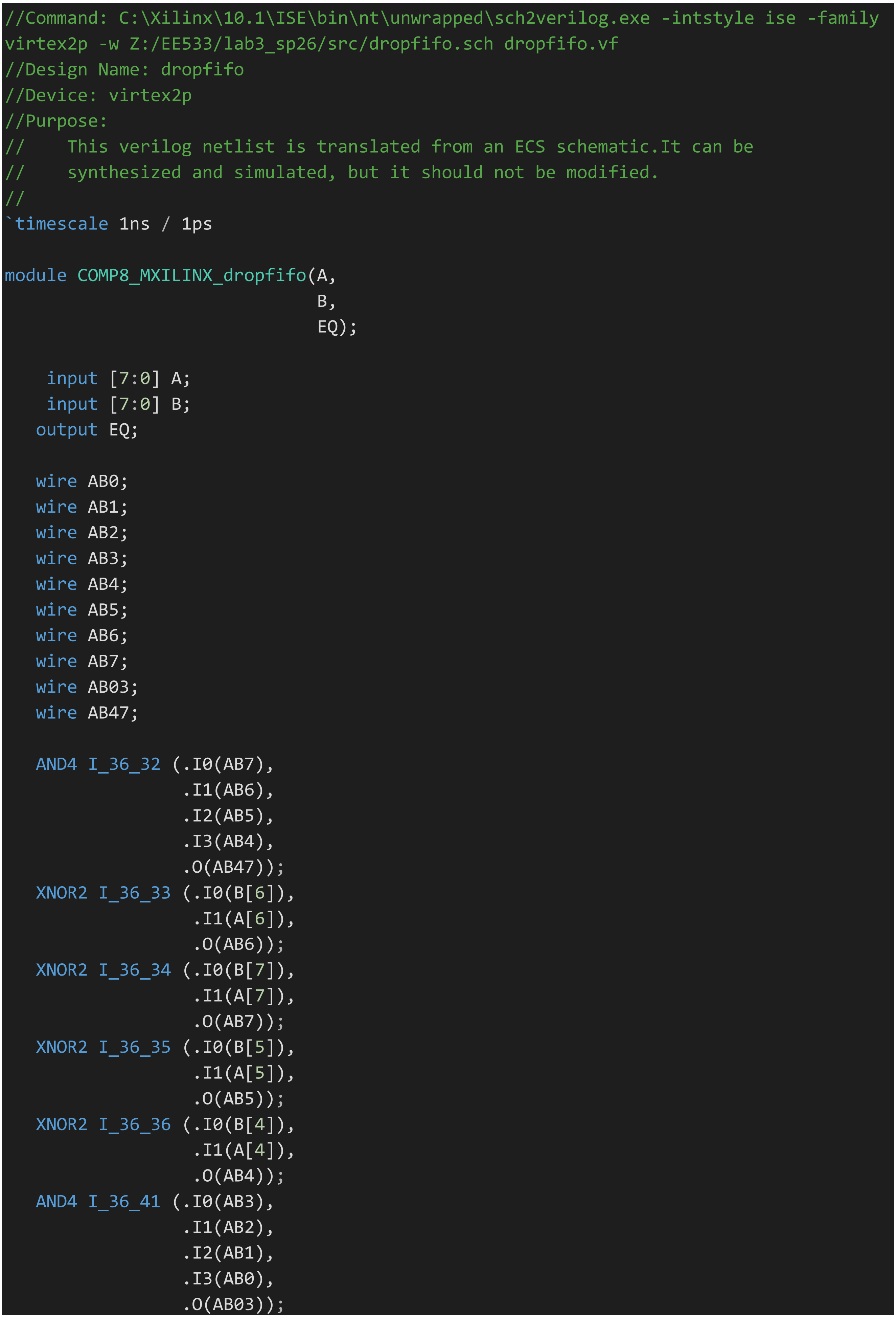


```
        .c_ywea_is_high(1),
        .c_yweb_is_high(1),
        .c_yydisable_warnings(1))
    inst (
        .ADDRA(addra),
        .ADDRB(addrb),
        .CLKA(clka),
        .CLKB(clkb),
        .DINA(dina),
        .DOUTB(doutb),
        .WEA(wea),
        .DINB(),
        .DOUTA(),
        .ENA(),
        .ENB(),
        .NDA(),
        .NDB(),
        .RFDA(),
        .RFDB(),
        .RDYA(),
        .RDYB(),
        .SINITA(),
        .SINITB(),
        .WEB());


// synthesis translate_on

// XST black box declaration
// box_type "black_box"
// synthesis attribute box_type of dual_port_mem is "black_box"

endmodule
```

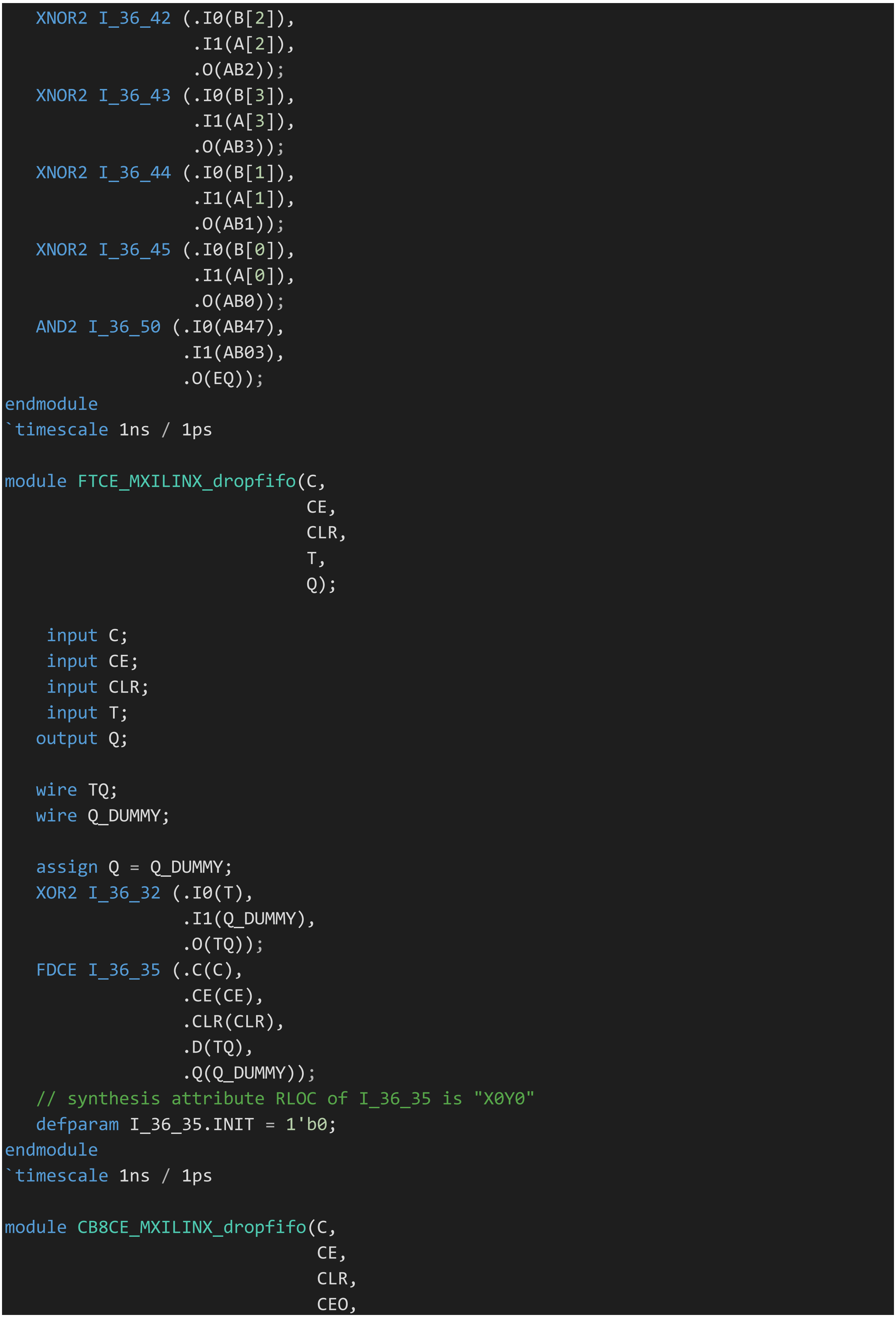
7. dropfifo.v



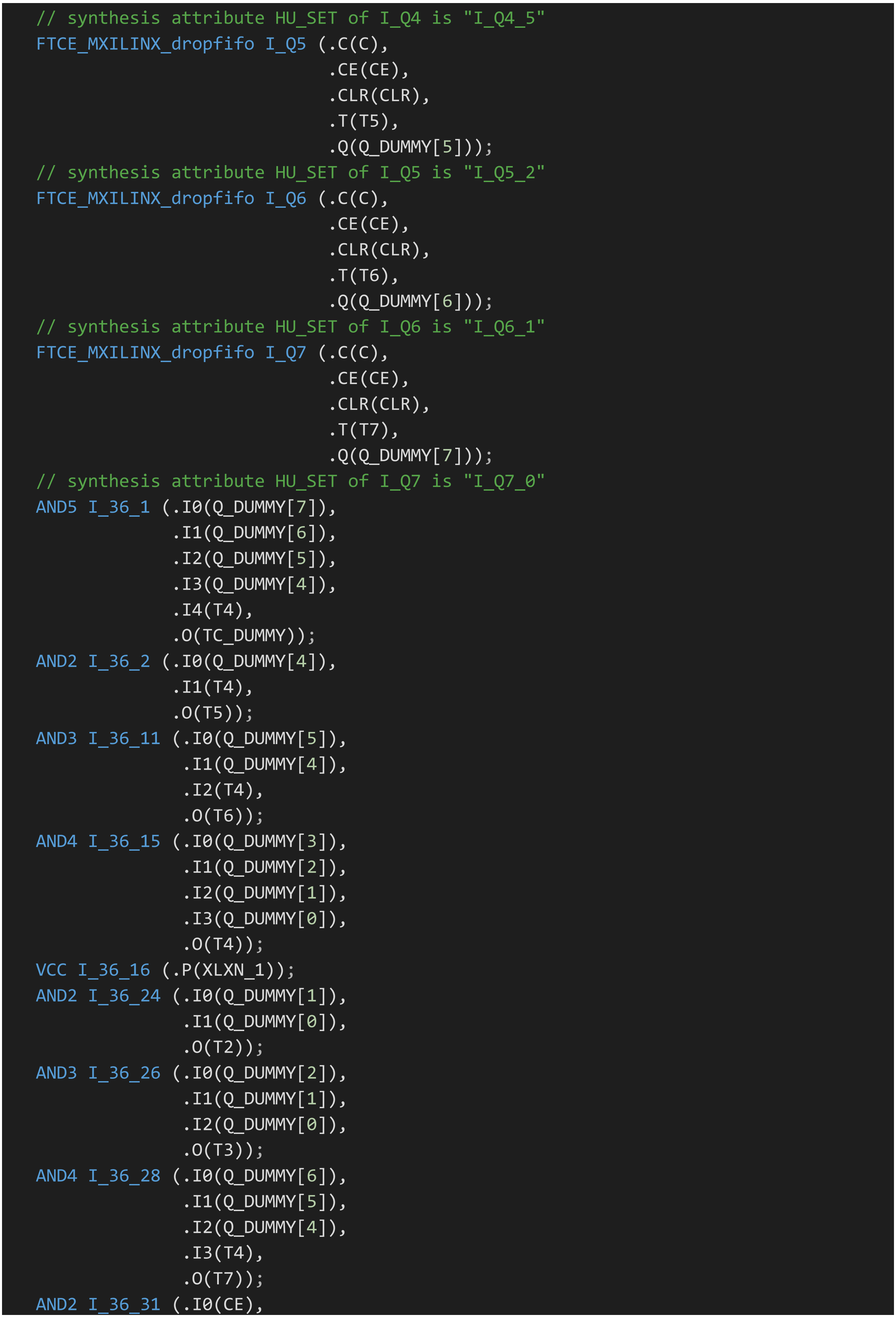
```
////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : dropfifo.vf
// /___/   /\     Timestamp : 01/30/2026 23:32:41
// \   \  /  \
//  \___\/\___\
//
```



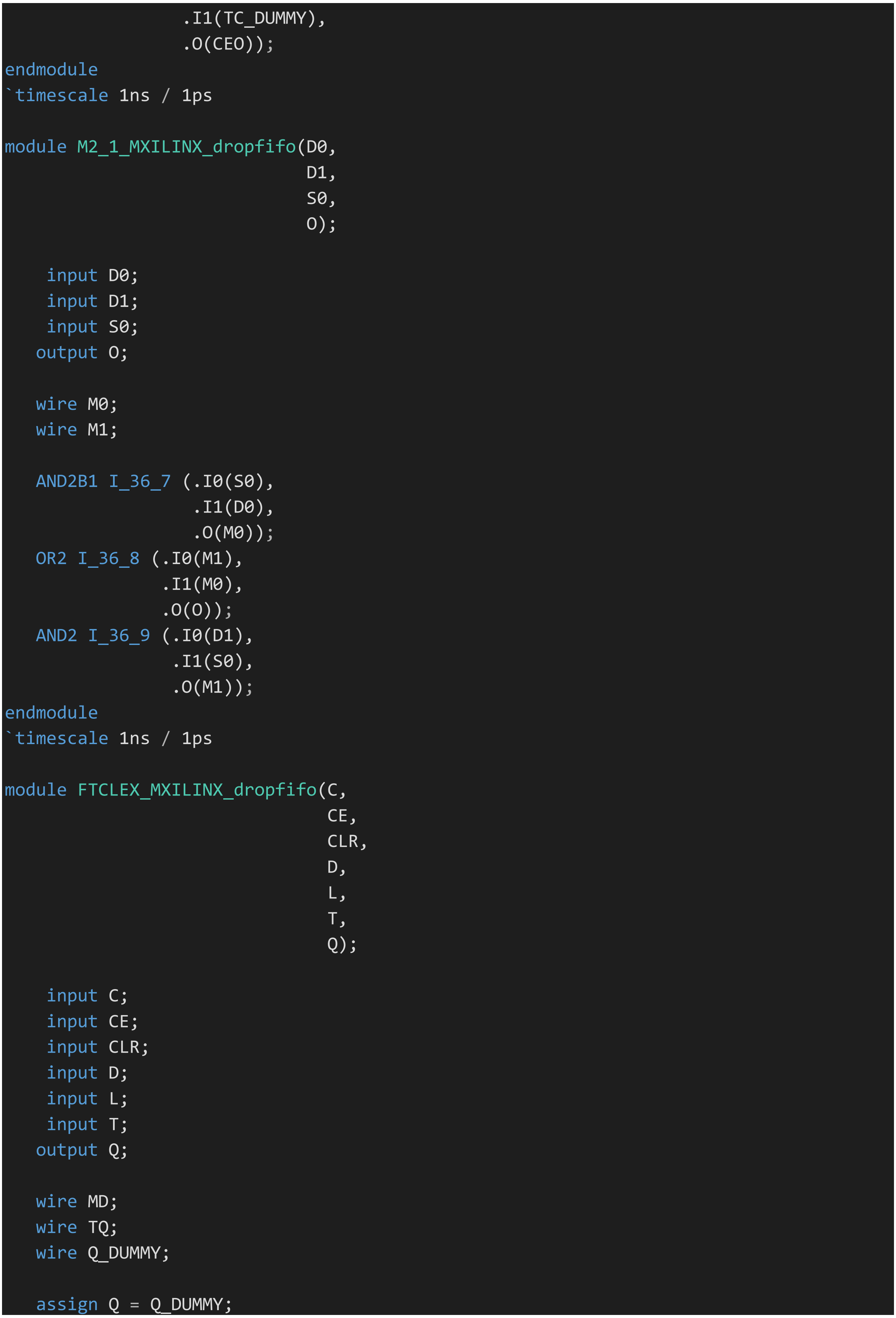
```
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family
virtex2p -w Z:/EE533/lab3_sp26/src/dropfifo.sch dropfifo.vf
//Design Name: dropfifo
//Device: virtex2p
//Purpose:
//    This verilog netlist is translated from an ECS schematic.It can be
//    synthesized and simulated, but it should not be modified.
//
`timescale 1ns / 1ps

module COMP8_MXILINX_dropfifo(A,
                              B,
                              EQ);

    input [7:0] A;
    input [7:0] B;
   output EQ;

   wire AB0;
   wire AB1;
   wire AB2;
   wire AB3;
   wire AB4;
   wire AB5;
   wire AB6;
   wire AB7;
   wire AB03;
   wire AB47;

   AND4 I_36_32 (.I0(AB7),
                 .I1(AB6),
                 .I2(AB5),
                 .I3(AB4),
                 .O(AB47));
   XNOR2 I_36_33 (.I0(B[6]),
                  .I1(A[6]),
                  .O(AB6));
   XNOR2 I_36_34 (.I0(B[7]),
                  .I1(A[7]),
                  .O(AB7));
   XNOR2 I_36_35 (.I0(B[5]),
                  .I1(A[5]),
                  .O(AB5));
   XNOR2 I_36_36 (.I0(B[4]),
                  .I1(A[4]),
                  .O(AB4));
   AND4 I_36_41 (.I0(AB3),
                 .I1(AB2),
                 .I2(AB1),
                 .I3(AB0),
                 .O(AB03));
```



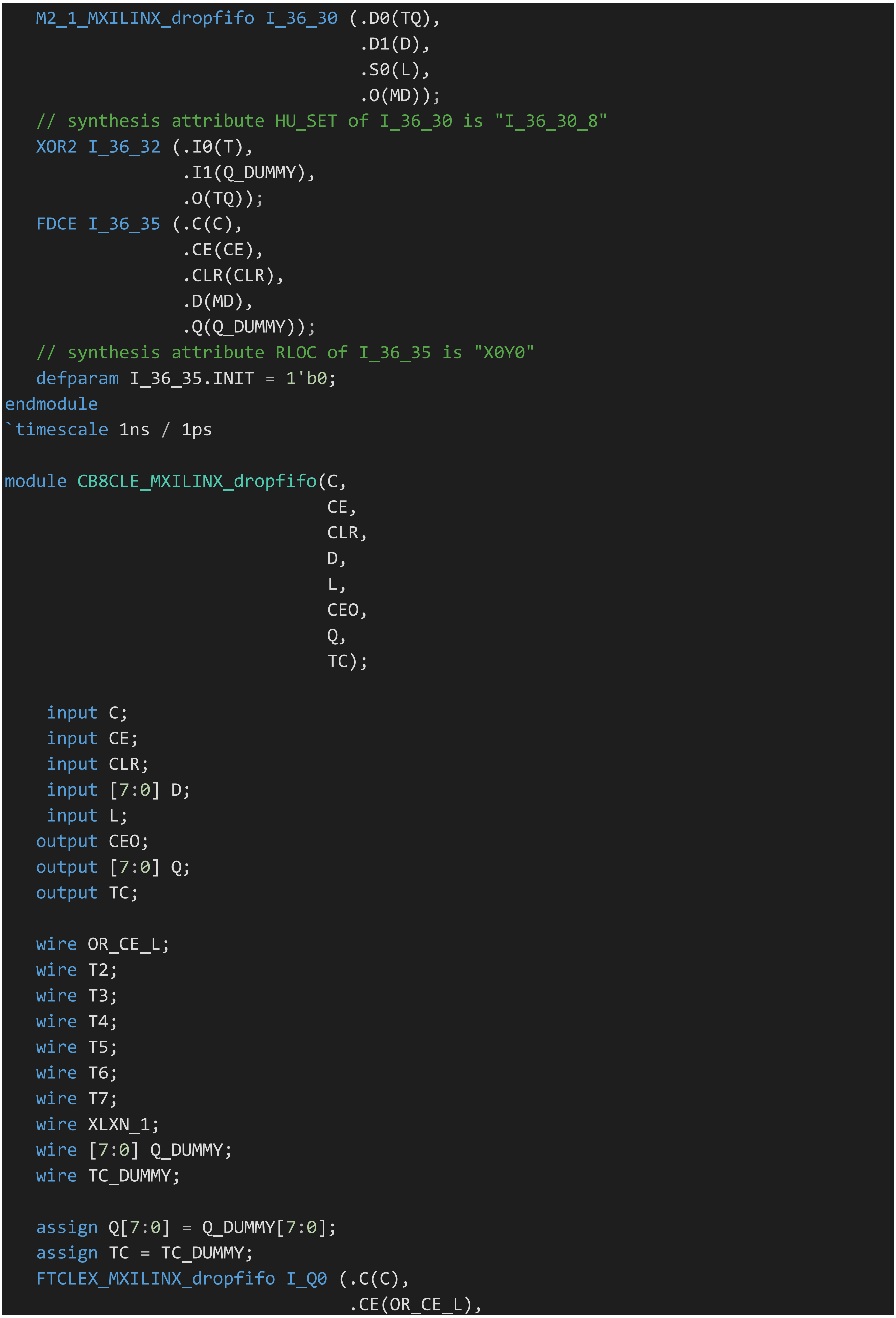
```
   XNOR2 I_36_42 (.I0(B[2]), 
                  .I1(A[2]), 
                  .O(AB2));
   XNOR2 I_36_43 (.I0(B[3]), 
                  .I1(A[3]), 
                  .O(AB3));
   XNOR2 I_36_44 (.I0(B[1]), 
                  .I1(A[1]), 
                  .O(AB1));
   XNOR2 I_36_45 (.I0(B[0]), 
                  .I1(A[0]), 
                  .O(AB0));
   AND2 I_36_50 (.I0(AB47), 
                 .I1(AB03), 
                 .O(EQ));
endmodule
`timescale 1ns / 1ps

module FTCE_MXILINX_dropfifo(C, 
                             CE, 
                             CLR, 
                             T, 
                             Q);

    input C;
    input CE;
    input CLR;
    input T;
   output Q;

   wire TQ;
   wire Q_DUMMY;

   assign Q = Q_DUMMY;
   XOR2 I_36_32 (.I0(T), 
                 .I1(Q_DUMMY), 
                 .O(TQ));
   FDCE I_36_35 (.C(C), 
                 .CE(CE), 
                 .CLR(CLR), 
                 .D(TQ), 
                 .Q(Q_DUMMY));
   // synthesis attribute RLOC of I_36_35 is "X0Y0"
   defparam I_36_35.INIT = 1'b0;
endmodule
`timescale 1ns / 1ps

module CB8CE_MXILINX_dropfifo(C, 
                              CE, 
                              CLR, 
                              CEO, 
                              Q,
                              TC);

    input C;
    input CE;
    input CLR;
   output CEO;
   output [7:0] Q;
   output TC;

   wire T2;
   wire T3;
   wire T4;
   wire T5;
   wire T6;
   wire T7;
   wire XLXN_1;
   wire [7:0] Q_DUMMY;
   wire TC_DUMMY;

   assign Q[7:0] = Q_DUMMY[7:0];
   assign TC = TC_DUMMY;
   FTCE_MXILINX_dropfifo I_Q0 (.C(C),
                               .CE(CE),
                               .CLR(CLR),
                               .T(XLXN_1),
                               .Q(Q_DUMMY[0]));
   // synthesis attribute HU_SET of I_Q0 is "I_Q0_6"
   FTCE_MXILINX_dropfifo I_Q1 (.C(C),
                               .CE(CE),
                               .CLR(CLR),
                               .T(Q_DUMMY[0]),
                               .Q(Q_DUMMY[1]));
   // synthesis attribute HU_SET of I_Q1 is "I_Q1_7"
   FTCE_MXILINX_dropfifo I_Q2 (.C(C),
                               .CE(CE),
                               .CLR(CLR),
                               .T(T2),
                               .Q(Q_DUMMY[2]));
   // synthesis attribute HU_SET of I_Q2 is "I_Q2_3"
   FTCE_MXILINX_dropfifo I_Q3 (.C(C),
                               .CE(CE),
                               .CLR(CLR),
                               .T(T3),
                               .Q(Q_DUMMY[3]));
   // synthesis attribute HU_SET of I_Q3 is "I_Q3_4"
   FTCE_MXILINX_dropfifo I_Q4 (.C(C),
                               .CE(CE),
                               .CLR(CLR),
                               .T(T4),
                               .Q(Q_DUMMY[4]));
```



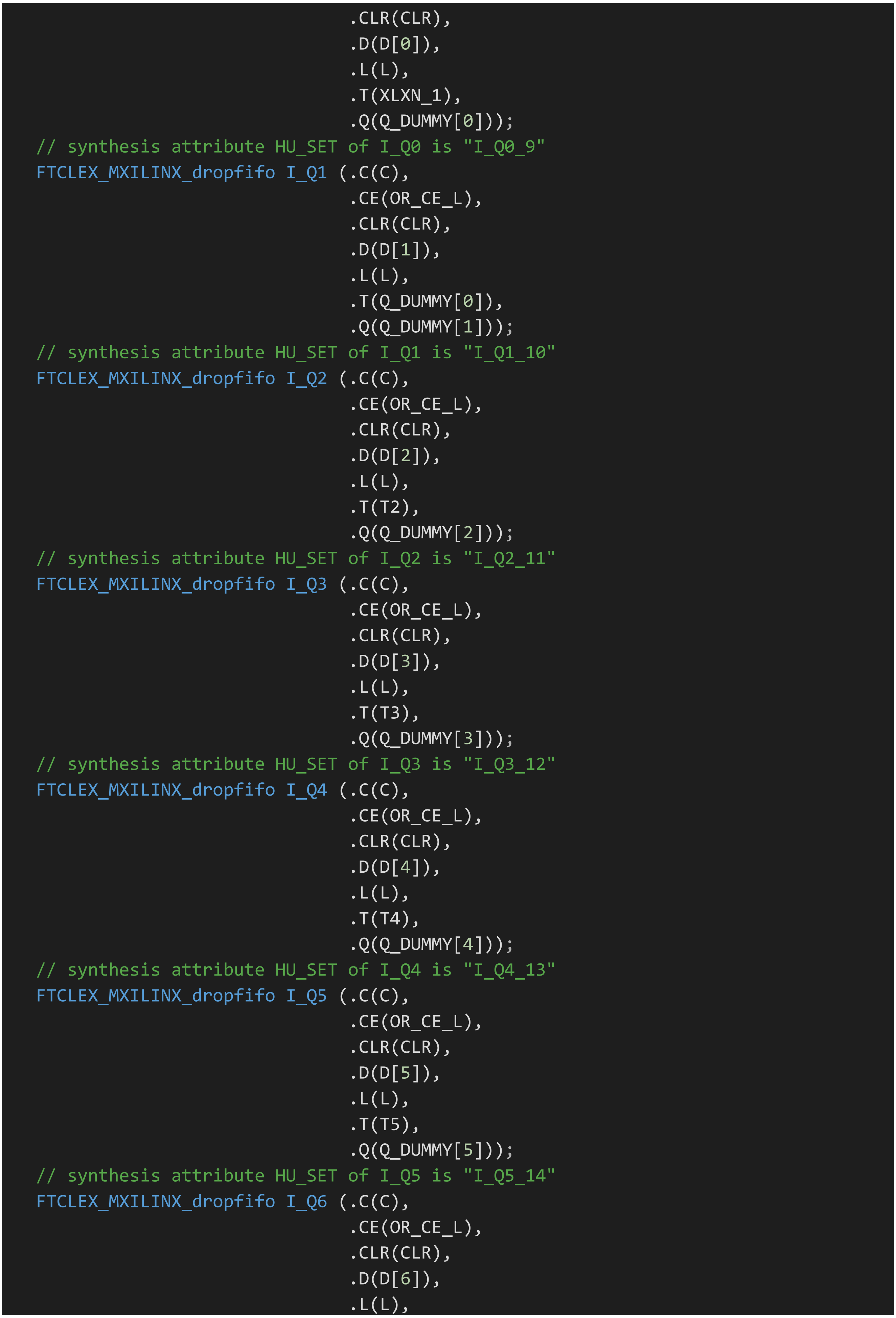
```
// synthesis attribute HU_SET of I_Q4 is "I_Q4_5"
FTCE_MXILINX_dropfifo I_Q5 (.C(C),
                            .CE(CE),
                            .CLR(CLR),
                            .T(T5),
                            .Q(Q_DUMMY[5]));
// synthesis attribute HU_SET of I_Q5 is "I_Q5_2"
FTCE_MXILINX_dropfifo I_Q6 (.C(C),
                            .CE(CE),
                            .CLR(CLR),
                            .T(T6),
                            .Q(Q_DUMMY[6]));
// synthesis attribute HU_SET of I_Q6 is "I_Q6_1"
FTCE_MXILINX_dropfifo I_Q7 (.C(C),
                            .CE(CE),
                            .CLR(CLR),
                            .T(T7),
                            .Q(Q_DUMMY[7]));
// synthesis attribute HU_SET of I_Q7 is "I_Q7_0"
AND5 I_36_1 (.I0(Q_DUMMY[7]),
             .I1(Q_DUMMY[6]),
             .I2(Q_DUMMY[5]),
             .I3(Q_DUMMY[4]),
             .I4(T4),
             .O(TC_DUMMY));
AND2 I_36_2 (.I0(Q_DUMMY[4]),
             .I1(T4),
             .O(T5));
AND3 I_36_11 (.I0(Q_DUMMY[5]),
              .I1(Q_DUMMY[4]),
              .I2(T4),
              .O(T6));
AND4 I_36_15 (.I0(Q_DUMMY[3]),
              .I1(Q_DUMMY[2]),
              .I2(Q_DUMMY[1]),
              .I3(Q_DUMMY[0]),
              .O(T4));
VCC I_36_16 (.P(XLXN_1));
AND2 I_36_24 (.I0(Q_DUMMY[1]),
              .I1(Q_DUMMY[0]),
              .O(T2));
AND3 I_36_26 (.I0(Q_DUMMY[2]),
              .I1(Q_DUMMY[1]),
              .I2(Q_DUMMY[0]),
              .O(T3));
AND4 I_36_28 (.I0(Q_DUMMY[6]),
              .I1(Q_DUMMY[5]),
              .I2(Q_DUMMY[4]),
              .I3(T4),
              .O(T7));
AND2 I_36_31 (.I0(CE),
```



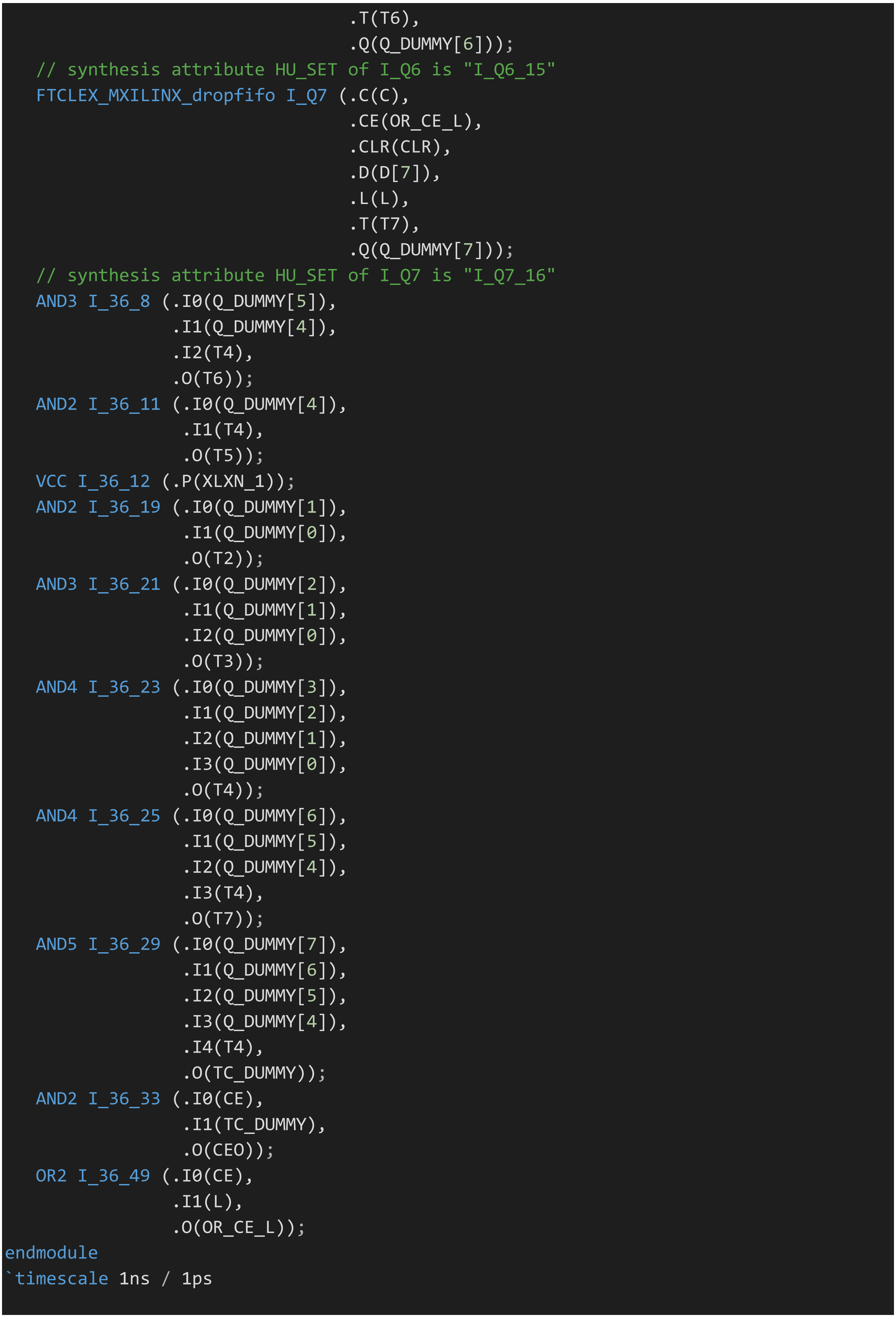
```
                 .I1(TC_DUMMY),
                 .O(CEO));
endmodule
`timescale 1ns / 1ps

module M2_1_MXILINX_dropfifo(D0,
                             D1,
                             S0,
                             O);

    input D0;
    input D1;
    input S0;
   output O;

   wire M0;
   wire M1;

   AND2B1 I_36_7 (.I0(S0),
                  .I1(D0),
                  .O(M0));
   OR2 I_36_8 (.I0(M1),
               .I1(M0),
               .O(O));
   AND2 I_36_9 (.I0(D1),
                .I1(S0),
                .O(M1));
endmodule
`timescale 1ns / 1ps

module FTCLEX_MXILINX_dropfifo(C,
                               CE,
                               CLR,
                               D,
                               L,
                               T,
                               Q);

    input C;
    input CE;
    input CLR;
    input D;
    input L;
    input T;
   output Q;

   wire MD;
   wire TQ;
   wire Q_DUMMY;

   assign Q = Q_DUMMY;
```



```
   M2_1_MXILINX_dropfifo I_36_30 (.D0(TQ),
                                  .D1(D),
                                  .S0(L),
                                  .O(MD));
   // synthesis attribute HU_SET of I_36_30 is "I_36_30_8"
   XOR2 I_36_32 (.I0(T),
                 .I1(Q_DUMMY),
                 .O(TQ));
   FDCE I_36_35 (.C(C),
                 .CE(CE),
                 .CLR(CLR),
                 .D(MD),
                 .Q(Q_DUMMY));
   // synthesis attribute RLOC of I_36_35 is "X0Y0"
   defparam I_36_35.INIT = 1'b0;
endmodule
`timescale 1ns / 1ps

module CB8CLE_MXILINX_dropfifo(C,
                               CE,
                               CLR,
                               D,
                               L,
                               CEO,
                               Q,
                               TC);

    input C;
    input CE;
    input CLR;
    input [7:0] D;
    input L;
   output CEO;
   output [7:0] Q;
   output TC;

   wire OR_CE_L;
   wire T2;
   wire T3;
   wire T4;
   wire T5;
   wire T6;
   wire T7;
   wire XLXN_1;
   wire [7:0] Q_DUMMY;
   wire TC_DUMMY;

   assign Q[7:0] = Q_DUMMY[7:0];
   assign TC = TC_DUMMY;
   FTCLEX_MXILINX_dropfifo I_Q0 (.C(C),
                                 .CE(OR_CE_L),
```

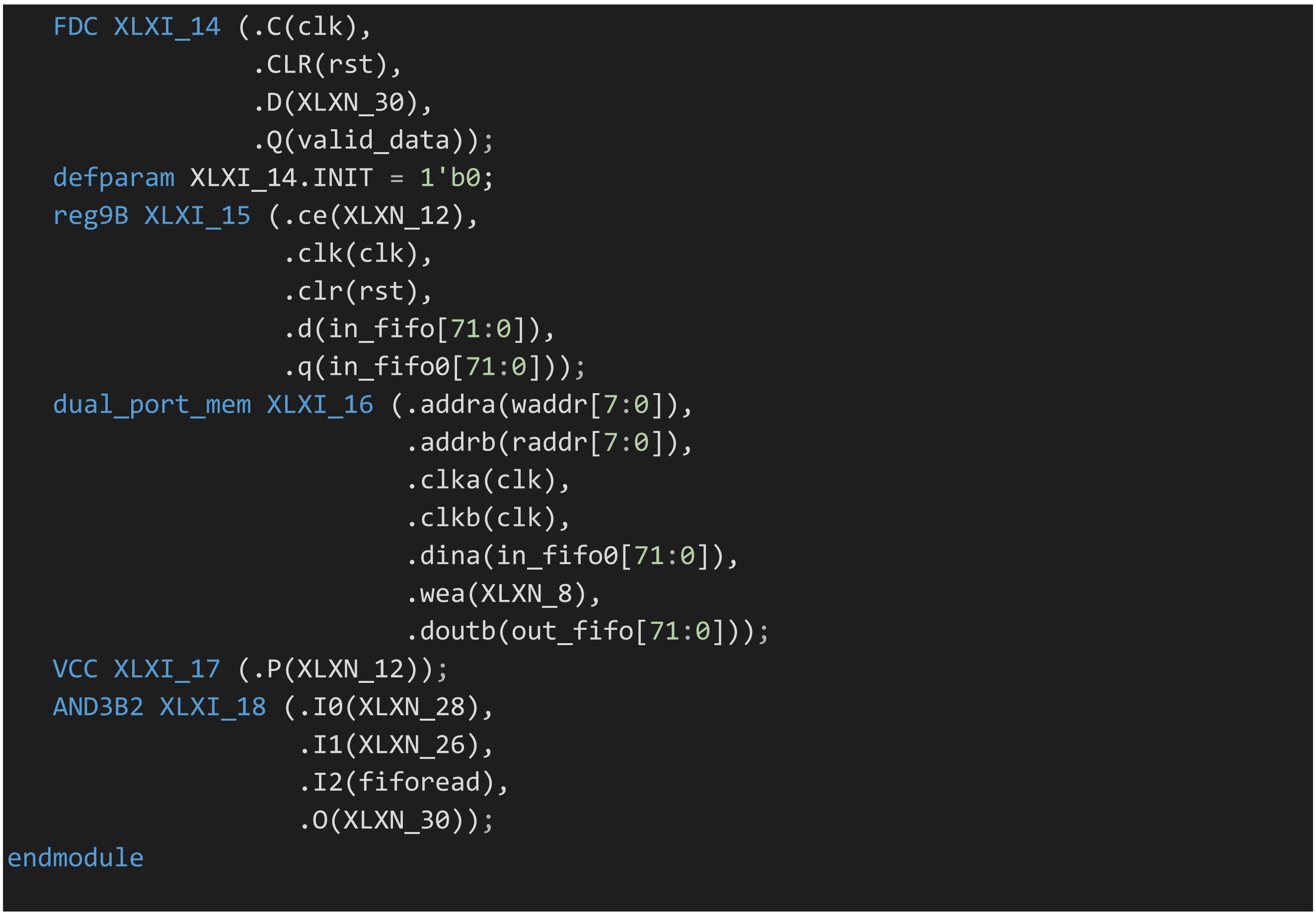


```
                              .CLR(CLR),
                              .D(D[0]),
                              .L(L),
                              .T(XLXN_1),
                              .Q(Q_DUMMY[0]));
// synthesis attribute HU_SET of I_Q0 is "I_Q0_9"
FTCLEX_MXILINX_dropfifo I_Q1 (.C(C),
                              .CE(OR_CE_L),
                              .CLR(CLR),
                              .D(D[1]),
                              .L(L),
                              .T(Q_DUMMY[0]),
                              .Q(Q_DUMMY[1]));
// synthesis attribute HU_SET of I_Q1 is "I_Q1_10"
FTCLEX_MXILINX_dropfifo I_Q2 (.C(C),
                              .CE(OR_CE_L),
                              .CLR(CLR),
                              .D(D[2]),
                              .L(L),
                              .T(T2),
                              .Q(Q_DUMMY[2]));
// synthesis attribute HU_SET of I_Q2 is "I_Q2_11"
FTCLEX_MXILINX_dropfifo I_Q3 (.C(C),
                              .CE(OR_CE_L),
                              .CLR(CLR),
                              .D(D[3]),
                              .L(L),
                              .T(T3),
                              .Q(Q_DUMMY[3]));
// synthesis attribute HU_SET of I_Q3 is "I_Q3_12"
FTCLEX_MXILINX_dropfifo I_Q4 (.C(C),
                              .CE(OR_CE_L),
                              .CLR(CLR),
                              .D(D[4]),
                              .L(L),
                              .T(T4),
                              .Q(Q_DUMMY[4]));
// synthesis attribute HU_SET of I_Q4 is "I_Q4_13"
FTCLEX_MXILINX_dropfifo I_Q5 (.C(C),
                              .CE(OR_CE_L),
                              .CLR(CLR),
                              .D(D[5]),
                              .L(L),
                              .T(T5),
                              .Q(Q_DUMMY[5]));
// synthesis attribute HU_SET of I_Q5 is "I_Q5_14"
FTCLEX_MXILINX_dropfifo I_Q6 (.C(C),
                              .CE(OR_CE_L),
                              .CLR(CLR),
                              .D(D[6]),
                              .L(L),
```



```
                                 .T(T6),
                                 .Q(Q_DUMMY[6]));
   // synthesis attribute HU_SET of I_Q6 is "I_Q6_15"
   FTCLEX_MXILINX_dropfifo I_Q7 (.C(C),
                                 .CE(OR_CE_L),
                                 .CLR(CLR),
                                 .D(D[7]),
                                 .L(L),
                                 .T(T7),
                                 .Q(Q_DUMMY[7]));
   // synthesis attribute HU_SET of I_Q7 is "I_Q7_16"
   AND3 I_36_8 (.I0(Q_DUMMY[5]),
                .I1(Q_DUMMY[4]),
                .I2(T4),
                .O(T6));
   AND2 I_36_11 (.I0(Q_DUMMY[4]),
                 .I1(T4),
                 .O(T5));
   VCC I_36_12 (.P(XLXN_1));
   AND2 I_36_19 (.I0(Q_DUMMY[1]),
                 .I1(Q_DUMMY[0]),
                 .O(T2));
   AND3 I_36_21 (.I0(Q_DUMMY[2]),
                 .I1(Q_DUMMY[1]),
                 .I2(Q_DUMMY[0]),
                 .O(T3));
   AND4 I_36_23 (.I0(Q_DUMMY[3]),
                 .I1(Q_DUMMY[2]),
                 .I2(Q_DUMMY[1]),
                 .I3(Q_DUMMY[0]),
                 .O(T4));
   AND4 I_36_25 (.I0(Q_DUMMY[6]),
                 .I1(Q_DUMMY[5]),
                 .I2(Q_DUMMY[4]),
                 .I3(T4),
                 .O(T7));
   AND5 I_36_29 (.I0(Q_DUMMY[7]),
                 .I1(Q_DUMMY[6]),
                 .I2(Q_DUMMY[5]),
                 .I3(Q_DUMMY[4]),
                 .I4(T4),
                 .O(TC_DUMMY));
   AND2 I_36_33 (.I0(CE),
                 .I1(TC_DUMMY),
                 .O(CEO));
   OR2 I_36_49 (.I0(CE),
                .I1(L),
                .O(OR_CE_L));
endmodule
`timescale 1ns / 1ps
```

```
module FD8CE_MXILINX_dropfifo(C,
                              CE,
                              CLR,
                              D,
                              Q);

    input C;
    input CE;
    input CLR;
    input [7:0] D;
   output [7:0] Q;


   FDCE I_Q0 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[0]),
              .Q(Q[0]));
   defparam I_Q0.INIT = 1'b0;
   FDCE I_Q1 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[1]),
              .Q(Q[1]));
   defparam I_Q1.INIT = 1'b0;
   FDCE I_Q2 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[2]),
              .Q(Q[2]));
   defparam I_Q2.INIT = 1'b0;
   FDCE I_Q3 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[3]),
              .Q(Q[3]));
   defparam I_Q3.INIT = 1'b0;
   FDCE I_Q4 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[4]),
              .Q(Q[4]));
   defparam I_Q4.INIT = 1'b0;
   FDCE I_Q5 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[5]),
              .Q(Q[5]));
   defparam I_Q5.INIT = 1'b0;
   FDCE I_Q6 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[6]),
              .Q(Q[6]));
   defparam I_Q6.INIT = 1'b0;
   FDCE I_Q7 (.C(C),
              .CE(CE),
              .CLR(CLR),
              .D(D[7]),
              .Q(Q[7]));
   defparam I_Q7.INIT = 1'b0;
endmodule
`timescale 1ns / 1ps

module dropfifo(clk,
                drop_pkt,
                fiforead,
                fifowrite,
                firstword,
                in_fifo,
                lastword,
                rst,
                out_fifo,
                valid_data);

    input clk;
    input drop_pkt;
    input fiforead;
    input fifowrite;
    input firstword;
    input [71:0] in_fifo;
    input lastword;
    input rst;
   output [71:0] out_fifo;
   output valid_data;

   wire [71:0] in_fifo0;
   wire [7:0] raddr;
   wire [7:0] waddr;
   wire XLXN_8;
   wire XLXN_12;
   wire [7:0] XLXN_15;
   wire XLXN_26;
   wire XLXN_28;
   wire XLXN_30;
   wire XLXN_35;
   wire XLXN_36;
   wire XLXN_37;
   wire XLXN_38;
   wire XLXN_40;

   FD XLXI_1 (.C(clk),
              .D(firstword),
              .Q(XLXN_38));
   defparam XLXI_1.INIT = 1'b0;
   FD XLXI_2 (.C(clk),
              .D(lastword),
              .Q(XLXN_37));
   defparam XLXI_2.INIT = 1'b0;
   FD8CE_MXILINX_dropfifo XLXI_3 (.C(clk),
                                  .CE(XLXN_40),
                                  .CLR(rst),
                                  .D(waddr[7:0]),
                                  .Q(XLXN_15[7:0]));
   // synthesis attribute HU_SET of XLXI_3 is "XLXI_3_17"
   OR2 XLXI_4 (.I0(XLXN_37),
               .I1(XLXN_38),
               .O(XLXN_36));
   AND2B1 XLXI_5 (.I0(XLXN_35),
                  .I1(XLXN_36),
                  .O(XLXN_40));
   FD XLXI_6 (.C(clk),
              .D(fifowrite),
              .Q(XLXN_8));
   defparam XLXI_6.INIT = 1'b0;
   COMP8_MXILINX_dropfifo XLXI_9 (.A(waddr[7:0]),
                                  .B(raddr[7:0]),
                                  .EQ(XLXN_26));
   // synthesis attribute HU_SET of XLXI_9 is "XLXI_9_20"
   COMP8_MXILINX_dropfifo XLXI_10 (.A(raddr[7:0]),
                                   .B(XLXN_15[7:0]),
                                   .EQ(XLXN_28));
   // synthesis attribute HU_SET of XLXI_10 is "XLXI_10_21"
   FD XLXI_11 (.C(clk),
               .D(drop_pkt),
               .Q(XLXN_35));
   defparam XLXI_11.INIT = 1'b0;
   CB8CLE_MXILINX_dropfifo XLXI_12 (.C(clk),
                                    .CE(XLXN_8),
                                    .CLR(rst),
                                    .D(XLXN_15[7:0]),
                                    .L(XLXN_35),
                                    .CEO(),
                                    .Q(waddr[7:0]),
                                    .TC());
   // synthesis attribute HU_SET of XLXI_12 is "XLXI_12_18"
   CB8CE_MXILINX_dropfifo XLXI_13 (.C(clk),
                                   .CE(XLXN_30),
                                   .CLR(rst),
                                   .CEO(),
                                   .Q(raddr[7:0]),
                                   .TC());
   // synthesis attribute HU_SET of XLXI_13 is "XLXI_13_19"
```



```
   FDC XLXI_14 (.C(clk),
                .CLR(rst),
                .D(XLXN_30),
                .Q(valid_data));
   defparam XLXI_14.INIT = 1'b0;
   reg9B XLXI_15 (.ce(XLXN_12),
                  .clk(clk),
                  .clr(rst),
                  .d(in_fifo[71:0]),
                  .q(in_fifo0[71:0]));
   dual_port_mem XLXI_16 (.addra(waddr[7:0]),
                          .addrb(raddr[7:0]),
                          .clka(clk),
                          .clkb(clk),
                          .dina(in_fifo0[71:0]),
                          .wea(XLXN_8),
                          .doutb(out_fifo[71:0]));
   VCC XLXI_17 (.P(XLXN_12));
   AND3B2 XLXI_18 (.I0(XLXN_28),
                   .I1(XLXN_26),
                   .I2(fiforead),
                   .O(XLXN_30));
endmodule
```