EE533 - Network Processor Design and Programming

Lab 3 Report: Mini-Intrusion Detection Engine Design

Student Name: Yanzhe Li Student ID: 6884027371
Email: yanzheli@usc.edu Date: January 31, 2026

1. Introduction

In this laboratory assignment, I designed and implemented a Mini-Intrusion Detection System (Mini-IDS) hardware engine. The design uses a combination of schematic capture, an IP Core (dual-port block memory), and Verilog. It is capable of scanning streaming network data for a specific 7-byte signature and dropping packets that contain the malicious pattern.

2. System Overview

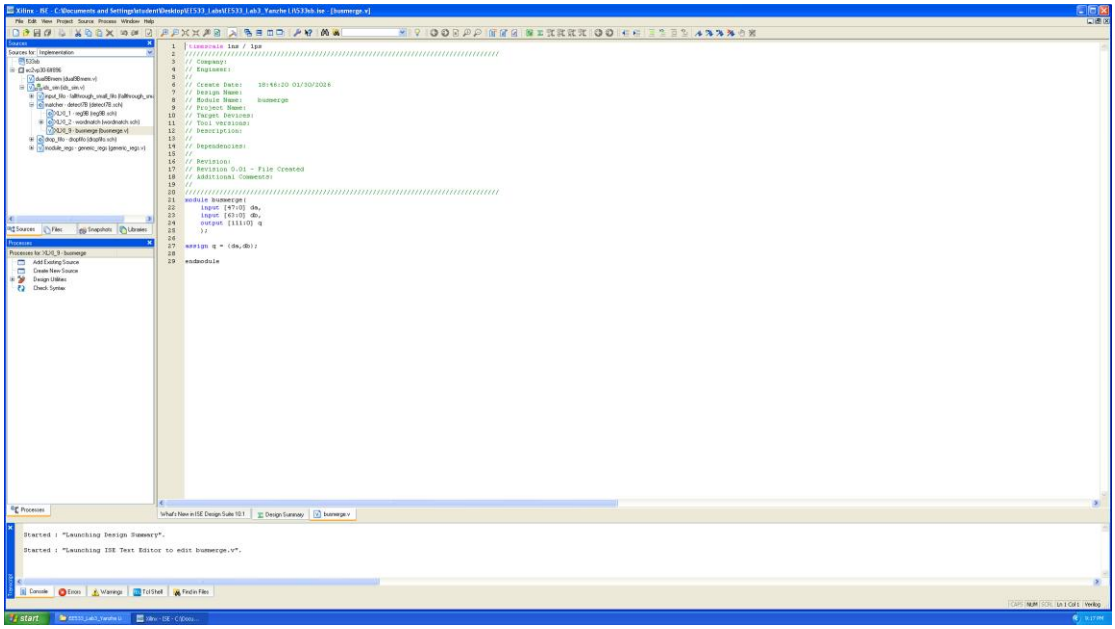
The Mini-IDS contains two main blocks:

1. Detection Pipeline: Builds a 112-bit window from consecutive data words and checks the 7-byte pattern at all byte alignments (0–7);
2. Drop FIFO Path: Buffers 72-bit words using dual-port RAM and controls forwarding vs. suppression based on detection results.

3. Module Summary

3.1 busmerge.v

Concatenates da[47:0] and db[63:0] into q[111:0] (q = {da, db}) to form the 14-byte sliding window.

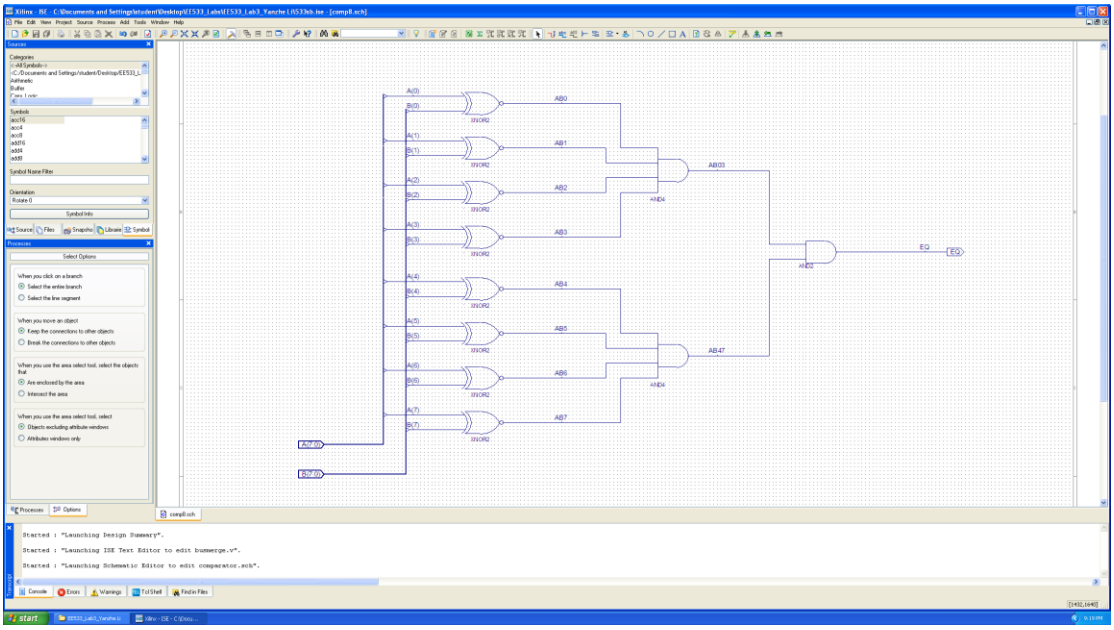


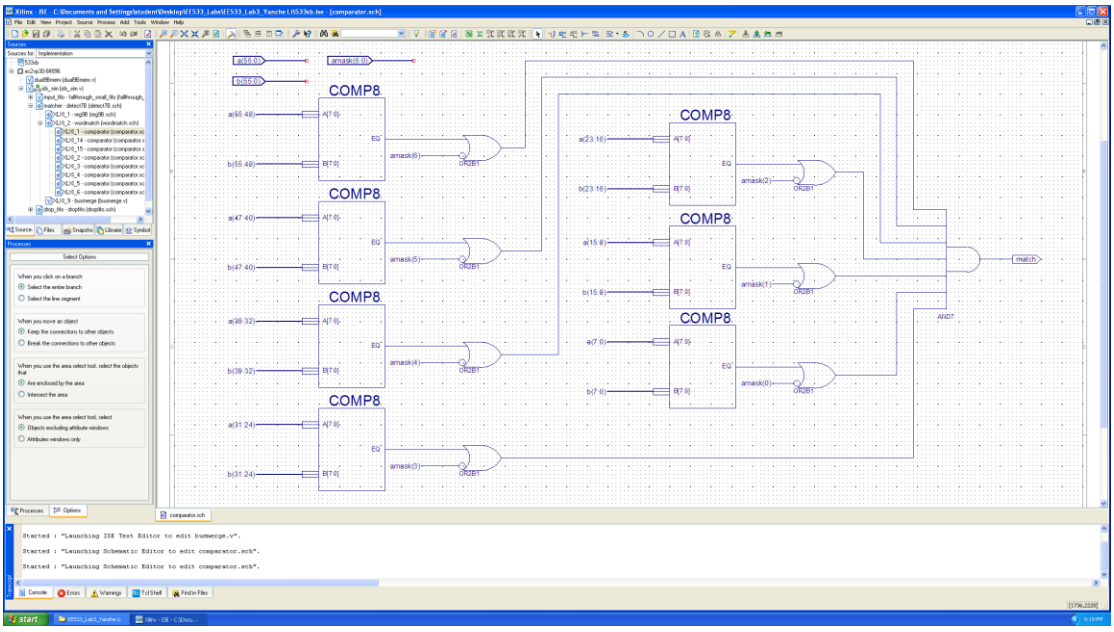
3.2 comp8.sch and comparator.sch

comp8.sch: 8-bit equality comparator (bitwise XNOR + AND);

comparator.sch: 7-byte comparator built from seven comp8 blocks;

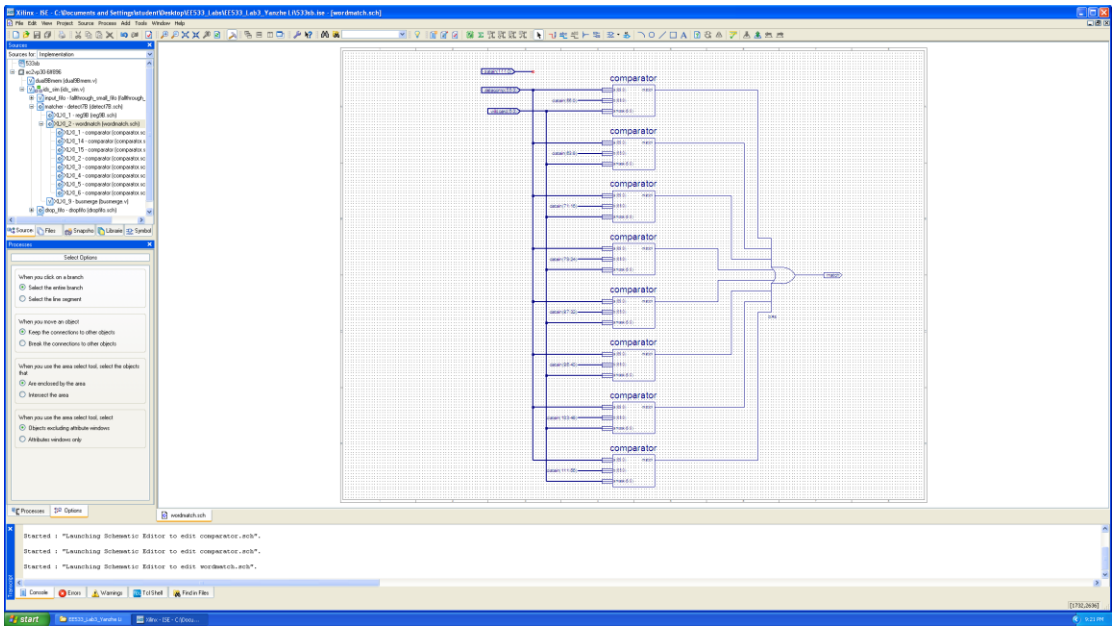
AMASK[6:0] provides per-byte wildcarding using OR2B1 gates.





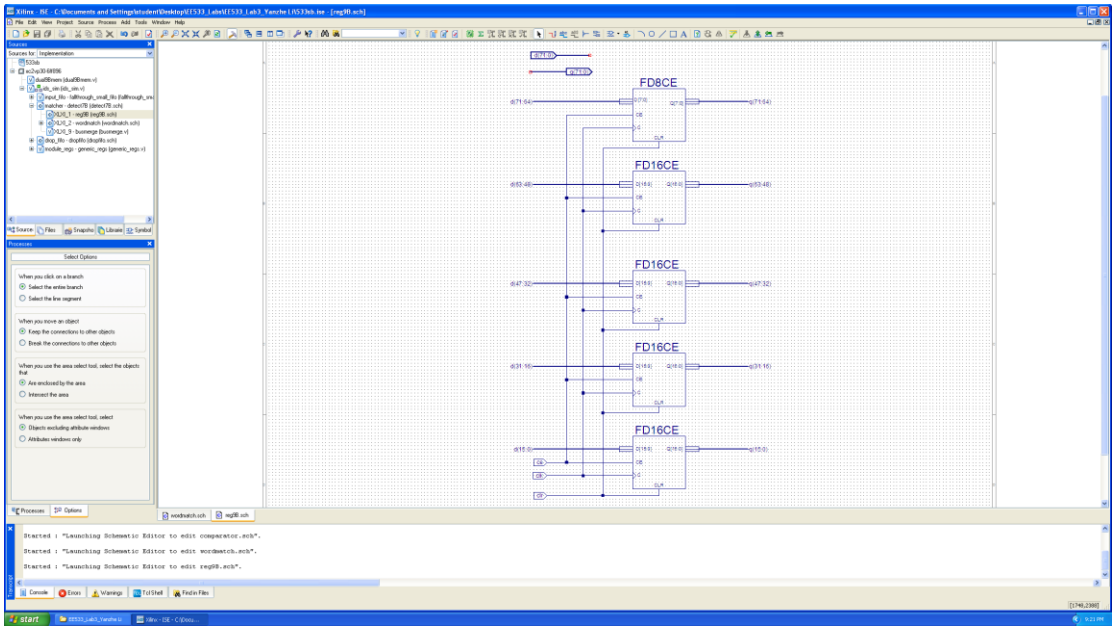
3.3 wordmatch.sch

It implements parallel matching across the 112-bit window by instantiating 8 comparator instances for byte offsets 0–7, and then OR'ing the results to produce a single match output.



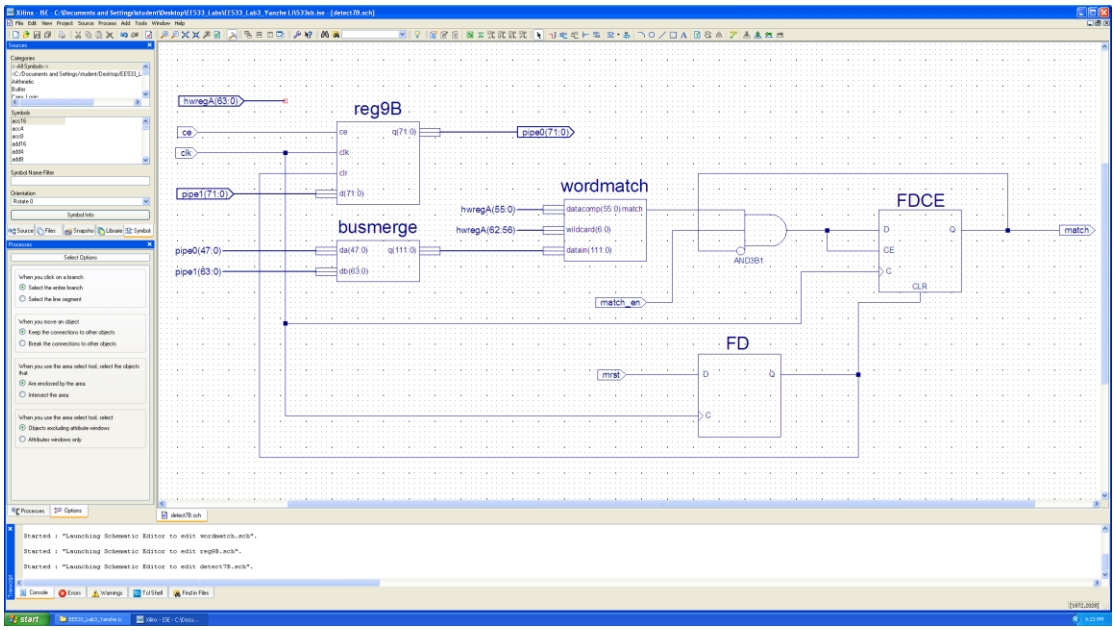
3.4 reg9B.sch

A 72-bit pipeline register is used for alignment and window construction, implemented using FD primitives.



3.5 detect7B.sch

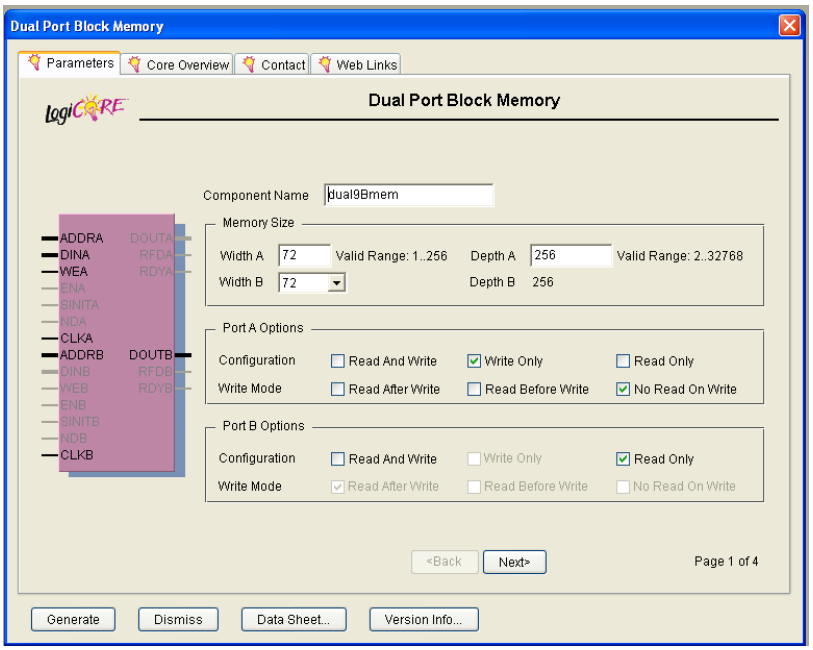
The top-level detection block integrates reg9B.sch, busmerge.sch, and wordmatch.sch. Detection is enabled by *match_en*, and the match result is registered for downstream control.

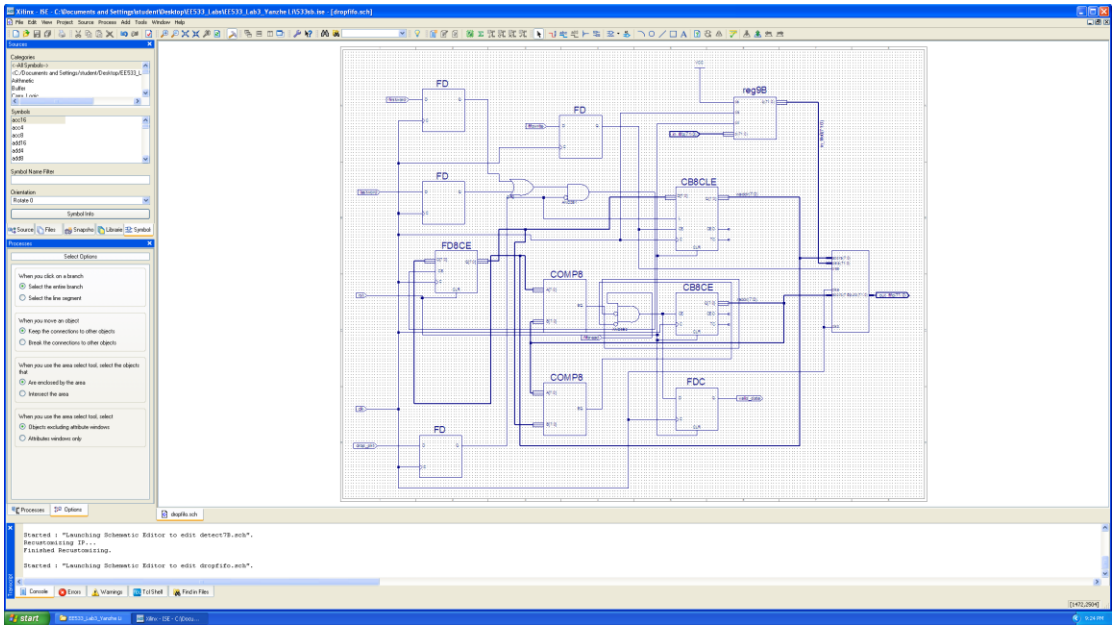


3.6 dropfifo.sch + dual9Bmem IP Core

IP Core: Simple dual-port RAM, Width = 72, Depth = 256.

It is used as FIFO storage to provide buffering and controlled output behavior when drops occur.





4. Pattern Matching Algorithm

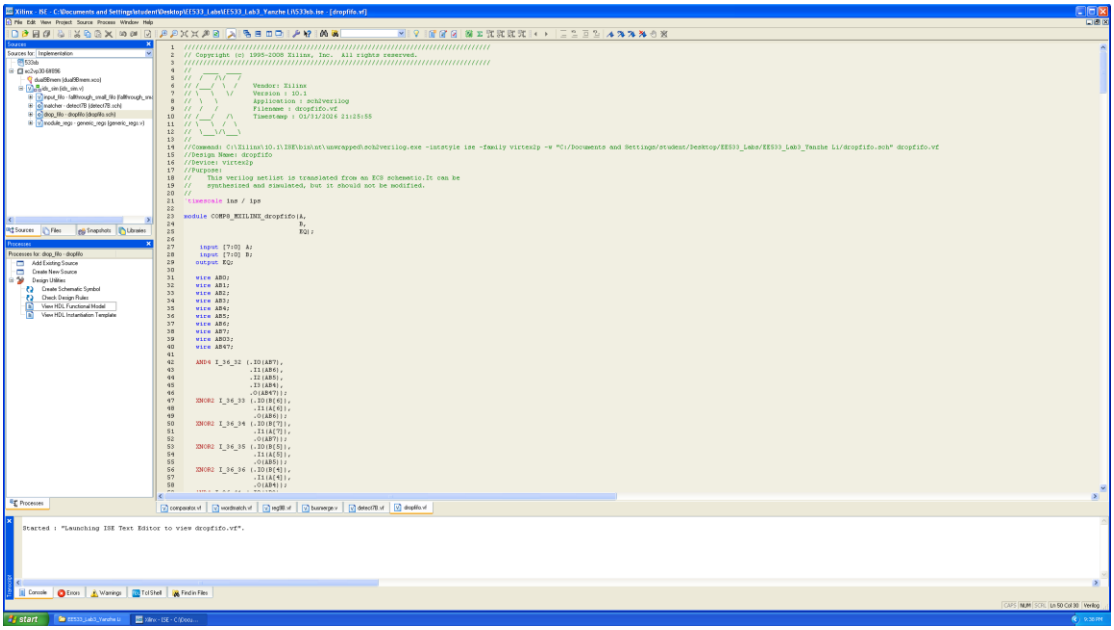
The Mini-IDS implements a byte-aligned sliding-window signature matching algorithm with per-byte wildcard support. It constructs a 112-bit (14-byte) window by combining overlapping portions of two consecutive 64-bit words, ensuring patterns that cross word boundaries can still be detected. A 56-bit (7-byte) signature is then compared against this window at all eight possible byte start offsets (0–7) in parallel. For each offset, the seven-byte comparisons are aggregated, while AMASK[6:0] enables “don’t care” behavior on selected bytes (masked bytes are treated as matching regardless of data; unmasked bytes must match exactly). The final match result is the OR of the eight alignment results, and the design can optionally register/latch the match under enable control to provide a stable detection signal for downstream logic.

5. Schematic-to-Verilog Conversion

All schematics were converted to structural Verilog using ISE utilities. The generated HDL matches the schematic connectivity and is suitable for synthesis and simulation.

Personal Evaluation (Schematic vs Verilog): Through this conversion process, I found that writing Verilog is generally easier and more efficient for complex logic, especially when dealing with repetitive structures like

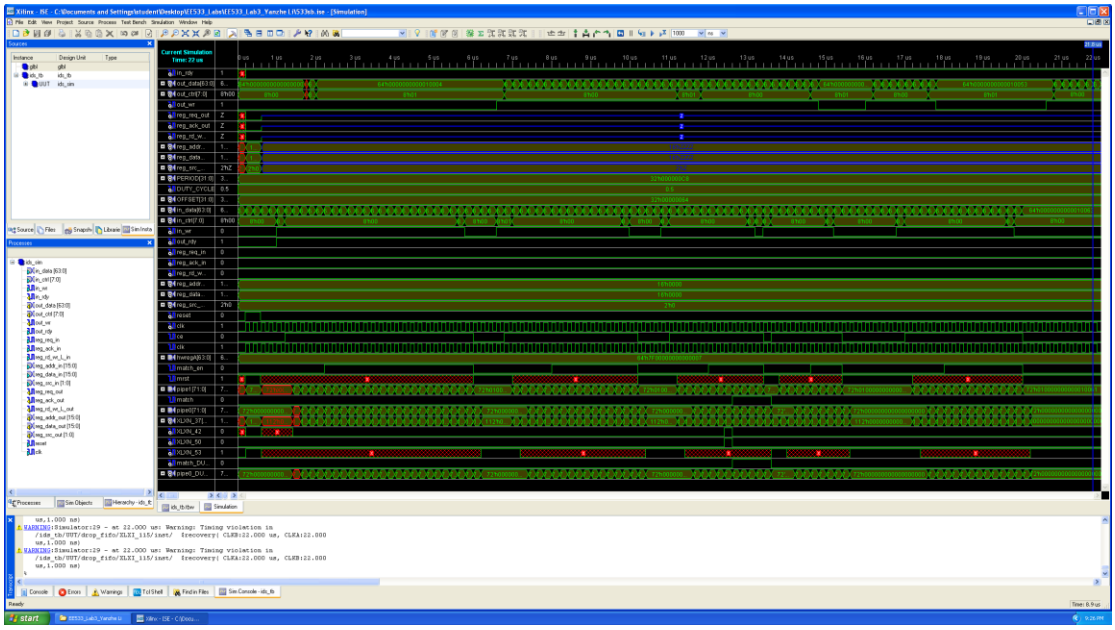
the eight parallel comparators in wordmatch.sch. In Verilog, this can be handled with simple loops or instantiation arrays. Whereas in Schematic Capture, it requires tedious manual wiring which is prone to connection errors. However, Schematics provided a better visual overview of the top-level data flow and module interconnects, making it easier to understand the system architecture at a glance.



6. Simulation Verification (ids_tb.tbw)-=09【

The provided *ids_tb.tbw* testbench emulates a NetFPGA-like streaming environment by generating packet-style input traffic and driving the Mini-IDS interface signals (clock/reset, data, and control).

During simulation, the incoming data stream continuously updates the pipeline registers and the 112-bit sliding window (datain[111:0]) used by the detector. The waveform shows that when the target 7-byte signature appears within this window, the internal pattern matcher asserts a brief hit pulse (XLXN_42, the wordmatch output). With detection enabled (match_en asserted), this hit produces a capture pulse (XLXN_50), allowing the registered detector output (match) to transition high. The fact that match is latched following the short wordmatch hit demonstrates that the detection pipeline correctly identifies the signature independent of byte alignment and provides a stable, registered indication suitable for downstream control.



7. Required Questions

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

AMASK[6:0] is a per-byte wildcard mask for the 7-byte signature comparison. In the schematic, each byte match result is combined with the mask using an OR2B1 gate (one inverted input), so the mask is effectively active-low: when AMASK[i]=0, that byte is treated as “don’t care” and always passes; when AMASK[i]=1, that byte must match exactly. This lets the IDS ignore certain bytes that may vary while still detecting the overall pattern.

7.2 What exactly does busmerge.v do?

busmerge.v simply concatenates two buses to build the 112-bit (14-byte) window: it combines da[47:0] (bytes kept from the previous cycle) with db[63:0] (the current 64-bit word) to produce q[111:0], i.e., q = {da, db}. This larger window helps detect a 7-byte signature even when it crosses the boundary between two 64-bit words.

7.3 What do the comp8 modules do in this schematic?

Each comp8 module compares two 8-bit values (one byte from data and one byte from the signature) and outputs EQ=1 only if they are exactly the same. It is built using bitwise XNOR logic followed by an AND so that all bits must match. These comp8 blocks are then used as the basic building

pieces for the full 7-byte comparison.

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

dual9Bmem is the dual-port block RAM used as the FIFO storage in dropfifo.sch. It stores 72-bit words (9 bytes) and is deep enough (256 entries) to buffer packet data. Because it is dual-port, the design can write incoming data and read outgoing data at the same time using separate addresses. This buffering allows the system to control output behavior, including suppressing data when a packet is flagged by the detection logic.

8. Appendix

Appendix A: Project Files Repository

GitHub Repository: https://github.com/guagua174/EE533-Lab3.git

Appendix B: Verilog Source Code

1. comparator.v

```
//////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
//////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : comparator.vf
// /___/   /\     Timestamp : 01/31/2026 19:41:57
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w
"C:/Documents and Settings/student/Desktop/533sb/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_12;
    wire XLXN_13;
    wire XLXN_16;
    wire XLXN_17;
    wire XLXN_19;
    wire XLXN_22;
    wire XLXN_24;
```

```
wire XLXN_85;
wire XLXN_86;
wire XLXN_87;
wire XLXN_88;
wire XLXN_89;
wire XLXN_132;

COMP8_MXILINX_comparator XLXI_1 (.A(a[55:48]),
                                 .B(b[55:48]),
                                 .EQ(XLXN_16));
// synthesis attribute HU_SET of XLXI_1 is "XLXI_1_0"
OR2B1 XLXI_2 (.I0(amask[6]),
              .I1(XLXN_16),
              .O(XLXN_8));
COMP8_MXILINX_comparator XLXI_3 (.A(a[31:24]),
                                 .B(b[31:24]),
                                 .EQ(XLXN_12));
// synthesis attribute HU_SET of XLXI_3 is "XLXI_3_1"
COMP8_MXILINX_comparator XLXI_4 (.A(a[47:40]),
                                 .B(b[47:40]),
                                 .EQ(XLXN_13));
// synthesis attribute HU_SET of XLXI_4 is "XLXI_4_2"
OR2B1 XLXI_5 (.I0(amask[5]),
              .I1(XLXN_13),
              .O(XLXN_85));
COMP8_MXILINX_comparator XLXI_6 (.A(a[39:32]),
                                 .B(b[39:32]),
                                 .EQ(XLXN_17));
// synthesis attribute HU_SET of XLXI_6 is "XLXI_6_3"
OR2B1 XLXI_7 (.I0(amask[4]),
              .I1(XLXN_17),
              .O(XLXN_86));
OR2B1 XLXI_8 (.I0(amask[3]),
```

```
                        .I1(XLXN_12),
                        .O(XLXN_132));
    COMP8_MXILINX_comparator XLXI_9 (.A(a[7:0]),
                                     .B(b[7:0]),
                                     .EQ(XLXN_24));
    // synthesis attribute HU_SET of XLXI_9 is "XLXI_9_4"
    COMP8_MXILINX_comparator XLXI_10 (.A(a[23:16]),
                                      .B(b[23:16]),
                                      .EQ(XLXN_19));
    // synthesis attribute HU_SET of XLXI_10 is "XLXI_10_5"
    COMP8_MXILINX_comparator XLXI_11 (.A(a[15:8]),
                                      .B(b[15:8]),
                                      .EQ(XLXN_22));
    // synthesis attribute HU_SET of XLXI_11 is "XLXI_11_6"
    OR2B1 XLXI_12 (.I0(amask[2]),
                   .I1(XLXN_19),
                   .O(XLXN_87));
    OR2B1 XLXI_13 (.I0(amask[1]),
                   .I1(XLXN_22),
                   .O(XLXN_88));
    OR2B1 XLXI_14 (.I0(amask[0]),
                   .I1(XLXN_24),
                   .O(XLXN_89));
    AND7_MXILINX_comparator XLXI_16 (.I0(XLXN_132),
                                     .I1(XLXN_89),
                                     .I2(XLXN_88),
                                     .I3(XLXN_87),
                                     .I4(XLXN_86),
                                     .I5(XLXN_85),
                                     .I6(XLXN_8),
                                     .O(match));
    // synthesis attribute HU_SET of XLXI_16 is "XLXI_16_7"
Endmodule
```

2. detect7B.v

```
//////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
//////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : detect7B.vf
// /___/   /\     Timestamp : 01/31/2026 19:41:58
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w "C:/Documents and Settings/student/Desktop/533sb/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,
                pipe0);
```

```
  input ce;
  input clk;
  input [63:0] hwregA;
  input match_en;
  input mrst;
  input [71:0] pipe1;
output match;
output [71:0] pipe0;

wire [111:0] XLXN_37;
wire XLXN_42;
wire XLXN_50;
wire XLXN_53;
wire match_DUMMY;
wire [71:0] pipe0_DUMMY;

assign match = match_DUMMY;
assign pipe0[71:0] = pipe0_DUMMY[71:0];
reg9B XLXI_1 (.ce(ce),
              .clk(clk),
              .clr(XLXN_53),
              .d(pipe1[71:0]),
              .q(pipe0_DUMMY[71:0]));
wordmatch XLXI_2 (.datacomp(hwregA[55:0]),
                  .datain(XLXN_37[111:0]),
                  .wildcard(hwregA[62:56]),
                  .match(XLXN_42));
FD XLXI_3 (.C(clk),
           .D(mrst),
           .Q(XLXN_53));
defparam XLXI_3.INIT = 1'b0;
AND3B1 XLXI_7 (.I0(match_DUMMY),
               .I1(match_en),
```

```
                        .I2(XLXN_42),
                        .O(XLXN_50));
    FDCE XLXI_8 (.C(clk),
                .CE(XLXN_50),
                .CLR(XLXN_53),
                .D(XLXN_50),
                .Q(match_DUMMY));
    defparam XLXI_8.INIT = 1'b0;
    busmerge XLXI_9 (.da(pipe0_DUMMY[47:0]),
                     .db(pipe1[63:0]),
                     .q(XLXN_37[111:0]));
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 21:09:18
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w "C:/Documents and Settings/student/Desktop/533sb/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. 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 21:09:19
// \     \   /   \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w "C:/Documents and Settings/student/Desktop/533sb/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_43;
  wire XLXN_80;
  wire XLXN_81;
  wire XLXN_82;
  wire XLXN_83;
  wire XLXN_84;
  wire XLXN_85;
  wire XLXN_86;

  comparator XLXI_1 (.a(datacomp[55:0]),
                     .amask(wildcard[6:0]),
                     .b(datain[55:0]),
                     .match(XLXN_43));
```

```
comparator XLXI_2 (.a(datacomp[55:0]),
                   .amask(wildcard[6:0]),
                   .b(datain[79:24]),
                   .match(XLXN_84));
comparator XLXI_3 (.a(datacomp[55:0]),
                   .amask(wildcard[6:0]),
                   .b(datain[87:32]),
                   .match(XLXN_83));
comparator XLXI_4 (.a(datacomp[55:0]),
                   .amask(wildcard[6:0]),
                   .b(datain[71:16]),
                   .match(XLXN_85));
comparator XLXI_5 (.a(datacomp[55:0]),
                   .amask(wildcard[6:0]),
                   .b(datain[63:8]),
                   .match(XLXN_86));
comparator XLXI_6 (.a(datacomp[55:0]),
                   .amask(wildcard[6:0]),
                   .b(datain[95:40]),
                   .match(XLXN_82));
comparator XLXI_14 (.a(datacomp[55:0]),
                    .amask(wildcard[6:0]),
                    .b(datain[103:48]),
                    .match(XLXN_81));
comparator XLXI_15 (.a(datacomp[55:0]),
                    .amask(wildcard[6:0]),
                    .b(datain[111:56]),
                    .match(XLXN_80));
OR8_MXILINX_wordmatch XLXI_18 (.I0(XLXN_80),
                               .I1(XLXN_81),
                               .I2(XLXN_82),
                               .I3(XLXN_83),
                               .I4(XLXN_84),
```

```
                                        .I5(XLXN_85),
                                        .I6(XLXN_86),
                                        .I7(XLXN_43),
                                        .O(match));
    // synthesis attribute HU_SET of XLXI_18 is "XLXI_18_0"
Endmodule
```

5. dropfifo.v

```
//////////////////////////////////////////////////////////////////////////////////
// Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.
//////////////////////////////////////////////////////////////////////////////////
//   ____  ____
//  /   /\/   /
// /___/  \  /    Vendor: Xilinx
// \   \   \/     Version : 10.1
//  \   \         Application : sch2verilog
//  /   /         Filename : dropfifo.vf
// /___/   /\     Timestamp : 01/31/2026 19:41:58
// \   \  /  \
//  \___\/\___\
//
//Command: C:\Xilinx\10.1\ISE\bin\nt\unwrapped\sch2verilog.exe -intstyle ise -family virtex2p -w
"C:/Documents and Settings/student/Desktop/533sb/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_1;
wire XLXN_2;
wire XLXN_3;
wire XLXN_11;
wire [7:0] XLXN_18;
wire XLXN_24;
wire XLXN_28;
wire XLXN_30;
wire XLXN_42;
wire XLXN_44;
wire XLXN_51;
```

```
FD XLXI_2 (.C(clk),
          .D(firstword),
          .Q(XLXN_1));
defparam XLXI_2.INIT = 1'b0;
FD XLXI_3 (.C(clk),
          .D(lastword),
          .Q(XLXN_2));
defparam XLXI_3.INIT = 1'b0;
FD XLXI_4 (.C(clk),
          .D(fifowrite),
          .Q(XLXN_11));
defparam XLXI_4.INIT = 1'b0;
FD8CE_MXILINX_dropfifo XLXI_5 (.C(clk),
                               .CE(XLXN_51),
                               .CLR(rst),
                               .D(waddr[7:0]),
                               .Q(XLXN_18[7:0]));
// synthesis attribute HU_SET of XLXI_5 is "XLXI_5_17"
FD XLXI_6 (.C(clk),
          .D(drop_pkt),
          .Q(XLXN_42));
defparam XLXI_6.INIT = 1'b0;
COMP8_MXILINX_dropfifo XLXI_7 (.A(waddr[7:0]),
                               .B(raddr[7:0]),
                               .EQ(XLXN_30));
// synthesis attribute HU_SET of XLXI_7 is "XLXI_7_21"
COMP8_MXILINX_dropfifo XLXI_8 (.A(raddr[7:0]),
                               .B(XLXN_18[7:0]),
                               .EQ(XLXN_28));
// synthesis attribute HU_SET of XLXI_8 is "XLXI_8_20"
OR2 XLXI_9 (.I0(XLXN_2),
            .I1(XLXN_1),
```

```
                .O(XLXN_3));
AND2B1 XLXI_10 (.I0(XLXN_42),
                .I1(XLXN_3),
                .O(XLXN_51));
CB8CLE_MXILINX_dropfifo XLXI_11 (.C(clk),
                                 .CE(XLXN_11),
                                 .CLR(rst),
                                 .D(XLXN_18[7:0]),
                                 .L(XLXN_42),
                                 .CEO(),
                                 .Q(waddr[7:0]),
                                 .TC());
// synthesis attribute HU_SET of XLXI_11 is "XLXI_11_18"
CB8CE_MXILINX_dropfifo XLXI_13 (.C(clk),
                                .CE(XLXN_24),
                                .CLR(rst),
                                .CEO(),
                                .Q(raddr[7:0]),
                                .TC());
// synthesis attribute HU_SET of XLXI_13 is "XLXI_13_19"
AND3B2 XLXI_14 (.I0(XLXN_28),
                .I1(XLXN_30),
                .I2(fiforead),
                .O(XLXN_24));
FDC XLXI_15 (.C(clk),
             .CLR(rst),
             .D(XLXN_24),
             .Q(valid_data));
defparam XLXI_15.INIT = 1'b0;
VCC XLXI_16 (.P(XLXN_44));
reg9B XLXI_112 (.ce(XLXN_44),
                .clk(clk),
                .clr(rst),
```

```
                                .d(in_fifo[71:0]),
                                .q(in_fifo0[71:0]));
    dual9Bmem XLXI_115 (.addra(waddr[7:0]),
                                .addrb(raddr[7:0]),
                                .clka(clk),
                                .clkb(clk),
                                .dina(in_fifo0[71:0]),
                                .wea(XLXN_11),
                                .doutb(out_fifo[71:0]));
endmodule
```