PARAMETERIZED ASYNCHRONOUS FIFO

INTRODUCTION

FIFOs are often used to safely pass data from one clock domain to another asynchronous clock domain. Using a FIFO to pass data from one clock domain to another clock domain requires multi-asynchronous clock design techniques.

Synchronous FIFO pointers

Synchronous FIFO pointers For synchronous FIFO design a FIFO where writes to, and reads from the FIFO buffer are conducted in the same clock domain), one implementation counts the number of writes to, and reads from the FIFO buffer to increment on FIFO write but no read , decrement on FIFO read but no write or hold no writes and reads, or simultaneous write and read operation the current fill value of the FIFO buffer. The FIFO is full when the FIFO counter reaches a predetermined full value and the FIFO is empty when the FIFO counter is zero.

Asynchronous FIFO pointer

An asynchronous FIFO refers to a FIFO design where data values are written to a FIFO buffer from one clock domain and the data values are read from the same FIFO buffer from another clock domain, where the two clock domains are asynchronous to each other

Working of Asynchronous FIFO

The write pointer always points to the next word to be written; therefore, on reset, both pointers are set to zero, which also happens to be the next FIFO word location to be written. On a FIFO-write operation, the memory location that is pointed to by the write pointer is written, and then the write pointer is incremented to point to the next location to be written. Similarly, the read pointer always points to the current FIFO word to be read. Again on reset, both pointers are reset to zero, the FIFO is empty and the read pointer is pointing to invalid data (because the FIFO is empty and the empty flag is asserted). As soon as the first data word is written to the FIFO, the write pointer increments, the empty flag is cleared, and the read pointer that is still addressing the contents of the first FIFO memory word, immediately drives that first valid word onto the FIFO data output port, to be read by the receiver logic.

OBJECTIVE

The objective of this project is to design and implement an asynchronous FIFO that safely transfers multi-bit data between two asynchronous clock domains. The FIFO uses Gray code pointers to enable reliable synchronization and accurate detection of full and empty conditions, avoiding metastability issues common in cross-domain data transfer. The design ensures safe pointer synchronization, correct full/empty flag generation, robust handling of overflow and underflow, and support for different clock speeds. It follows industry-standard techniques to make the FIFO synthesizable, verifiable, and timing-closure friendly, ensuring high reliability in real-world ASIC and FPGA applications.

Industry standard tools used: Cadence Genus Synthesis solution, Xilinx Vivado

CODE

Module 1: Asynchronous FIFO

```
module async_fifo1
    #(
      parameter DSIZE = 8,
      parameter ASIZE = 4
     )
    (
      input   winc, wclk, wrst_n,
      input   rinc, rclk, rrst_n,
      input   [DSIZE-1:0] wdata,

      output  [DSIZE-1:0] rdata,
      output  wfull,
      output  rempty
    );

      wire [ASIZE-1:0] waddr, raddr;
      initial $display("ASIZE IS %d", ASIZE);

      wire [ASIZE:0] wptr, rptr, wq2_rptr, rq2_wptr;

              sync_r2w sync_r2w (.wq2_rptr(wq2_rptr), .rptr(rptr),
.wclk(wclk), .wrst_n(wrst_n));
             sync_w2r sync_w2r (.rq2_wptr(rq2_wptr), .wptr(wptr),
.rclk(rclk), .rrst_n(rrst_n));
 fifomem #(DSIZE, ASIZE) fifomem
(.rclken(rinc),.rempty(rempty),.rdata(rdata),      .wdata(wdata),
.waddr(waddr), .raddr(raddr), .wclken(winc),
.wfull(wfull).wclk(wclk),.rclk(rclk));
 rptr_empty #(ASIZE,DSIZE) rptr_empty (.rempty(rempty),
.raddr(raddr), .rptr(rptr), .rq2_wptr(rq2_wptr), .rinc(rinc),
.rclk(rclk), .rrst_n(rrst_n),.rdata(rdata));
 wptr_full #(ASIZE) wptr_full (.wfull(wfull), .waddr(waddr),
.wptr(wptr), .wq2_rptr(wq2_rptr), .winc(winc), .wclk(wclk),
.wrst_n(wrst_n));
   endmodule
```

Module 2: FIFO Memory

```
module fifomem
#(
  parameter DATASIZE = 8,
  parameter ADDRSIZE = 3'b100
)
(
  input   wclken, wfull, wclk,rempty,rclk,rclken,
  input   [ADDRSIZE-1:0] waddr, raddr,
  input   [DATASIZE-1:0] wdata,
  output reg  [DATASIZE-1:0] rdata
);

  localparam DEPTH = 1<<(ADDRSIZE);
  initial
  $display("DEPTH IS %d", DEPTH);

  reg [DATASIZE-1:0] mem [0:DEPTH-1];
  always @(posedge wclk)
    if (wclken && !wfull)
      mem[waddr] <= wdata;

    always @(posedge rclk)
    if (rclken && !rempty)
      rdata <= mem[raddr];
endmodule
```

Module 3: Write Pointer Full Condition

```
module wptr_full
#(
  parameter ADDRSIZE = 4
)
(
  input   winc, wclk, wrst_n,
  input   [ADDRSIZE :0] wq2_rptr,
  output reg  wfull,
  output  [ADDRSIZE-1:0] waddr,
  output reg [ADDRSIZE :0] wptr
);
```

```
  reg [ADDRSIZE:0] wbin;
  wire [ADDRSIZE:0] wgraynext, wbinnext;
  wire wfull_val;

  always @(posedge wclk or negedge wrst_n)
    if (!wrst_n)
      {wbin, wptr} <= 0;
    else begin
         wbin <= wbinnext;
         wptr<= wgraynext;
         End

  assign wbinnext = wbin + (winc & ~wfull);
  assign waddr = wbin[ADDRSIZE-1:0];
  assign wgraynext = (wbinnext>>1) ^ wbinnext;

  assign wfull_val = (wgraynext=={~wq2_rptr[ADDRSIZE:ADDRSIZE-1],
wq2_rptr[ADDRSIZE-2:0]});

  always @(posedge wclk or negedge wrst_n)
    if (!wrst_n)
      wfull <= 1'b0;
    else
      wfull <= wfull_val;

endmodule
```

Module 4: Read Pointer Empty Condition

```
module rptr_empty
#(
  parameter ADDRSIZE = 4,
  parameter DSIZE=8
)
(
  input   rinc, rclk, rrst_n,
  input   [ADDRSIZE :0] rq2_wptr,
  output  reg  rempty,
  output  [ADDRSIZE-1:0] raddr,
  output [DSIZE-1:0] rdata,
  output reg [ADDRSIZE :0] rptr
```

```
);

  reg [ADDRSIZE:0] rbin;
  wire [ADDRSIZE:0] rgraynext, rbinnext;
  wire rempty_val;


  always @(posedge rclk or negedge rrst_n)
    if (!rrst_n)
      {rbin, rptr} <= 0;
    else
      {rbin, rptr} <= {rbinnext, rgraynext};

  assign rbinnext = rbin + (rinc & ~rempty);
  assign raddr = rbin[ADDRSIZE-1:0];
  assign rgraynext = (rbinnext>>1) ^ rbinnext;
  assign rempty_val = (rgraynext == rq2_wptr);

  always @(posedge rclk or negedge rrst_n)
    if (!rrst_n)
      rempty <= 1'b1;
    else
      rempty <= rempty_val;

endmodule
```

Module 5: Synchronisation From Read Pointer To Write Pointer

```
module sync_r2w
#(
  parameter ADDRSIZE = 4
)
(
  input   wclk, wrst_n,
  input   [ADDRSIZE:0] rptr,
  output reg  [ADDRSIZE:0] wq2_rptr
);

  reg [ADDRSIZE:0] wq1_rptr;

  always @(posedge wclk or negedge wrst_n)
    if (!wrst_n) {wq2_rptr,wq1_rptr} <= 0;
    else {wq2_rptr,wq1_rptr} <= {wq1_rptr,rptr};
```

```
endmodule
```

Module 6: Synchronization from Write Pointer to Read Pointer

```
module sync_r2w
#(
  parameter ADDRSIZE = 4
)
(
  input   wclk, wrst_n,
  input   [ADDRSIZE:0] rptr,
  output reg  [ADDRSIZE:0] wq2_rptr
);

  reg [ADDRSIZE:0] wq1_rptr;

  always @(posedge wclk or negedge wrst_n)
    if (!wrst_n) {wq2_rptr,wq1_rptr} <= 0;
    else {wq2_rptr,wq1_rptr} <= {wq1_rptr,rptr};

endmodule
```

TESTBENCH

```
module async_fifo_tb;

    parameter DSIZE = 8;
    parameter ASIZE = 4;
    parameter DEPTH = 1 << ASIZE;

    reg [DSIZE-1:0] wdata;
    wire [DSIZE-1:0] rdata;
    wire wfull, rempty;
    reg winc, rinc, wclk, rclk, wrst_n, rrst_n;

    async_fifo1 #(DSIZE, ASIZE) fifo (
        .rdata(rdata),
        .wdata(wdata),
        .wfull(wfull),
        .rempty(rempty),
        .winc(winc),
        .rinc(rinc),
```

```
    .wclk(wclk),
    .rclk(rclk),
    .wrst_n(wrst_n),
    .rrst_n(rrst_n)
);

always #5 wclk = ~wclk;
always #10 rclk = ~rclk;

initial begin

    wclk = 0;
    rclk = 0;
    wrst_n = 1;
    rrst_n=1;
    #1 wrst_n=0;
    rrst_n = 0;
    winc = 0;
    rinc = 0;

    #40 wrst_n = 1;winc=1;rrst_n = 1 ;
        wdata = 23;
    #10 wdata=43;
    #10 wdata=42;
    #10 wdata=44;
    #10 wdata=45;
    #10 wdata=47;
    #10 wdata=1;
    #10 wdata=3;
    #10 wdata=11;
    #10 wdata=4;
    #10 wdata=54;
    #10 wdata=65;
    #10 wdata=77;
    #10 wdata=89;
    #10 wdata=99;
    #10 wdata=121;
     #10 wdata=142;
    #10 wdata=44;
    #10 wdata=5;rinc=1;
    #10 wdata=7;
    #10 wdata=10;
    #10 wdata=13;
```

```
        #10 wdata=11;
        winc=0;

      #1000  $finish;
    end

endmodule
```

SIMULATION RESULTS

- Generated using Xilinx Vivado.

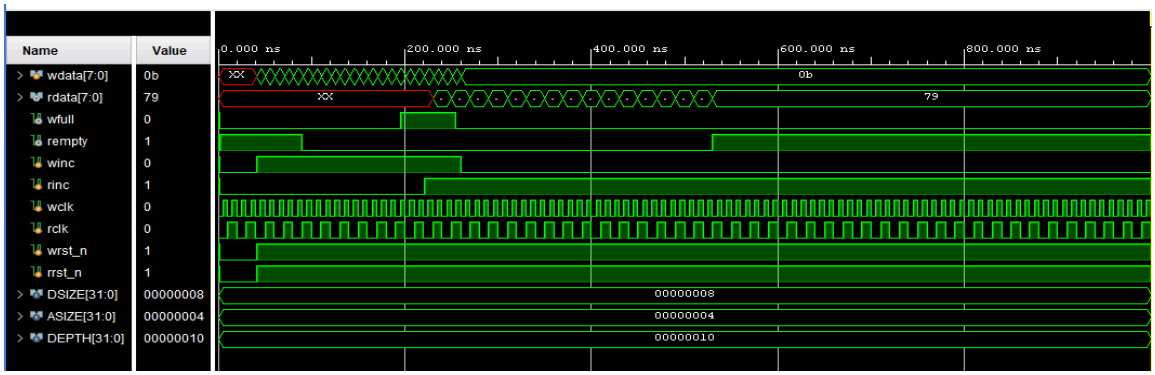

Figure 1. Xilinx Vivado Simulation Results

SYNTHESIS RESULTS:

- The reports below are generated using the Genus Synthesis Solution.

Module 1: Asynchronous FIFO

1. **Schematic after synthesis:**

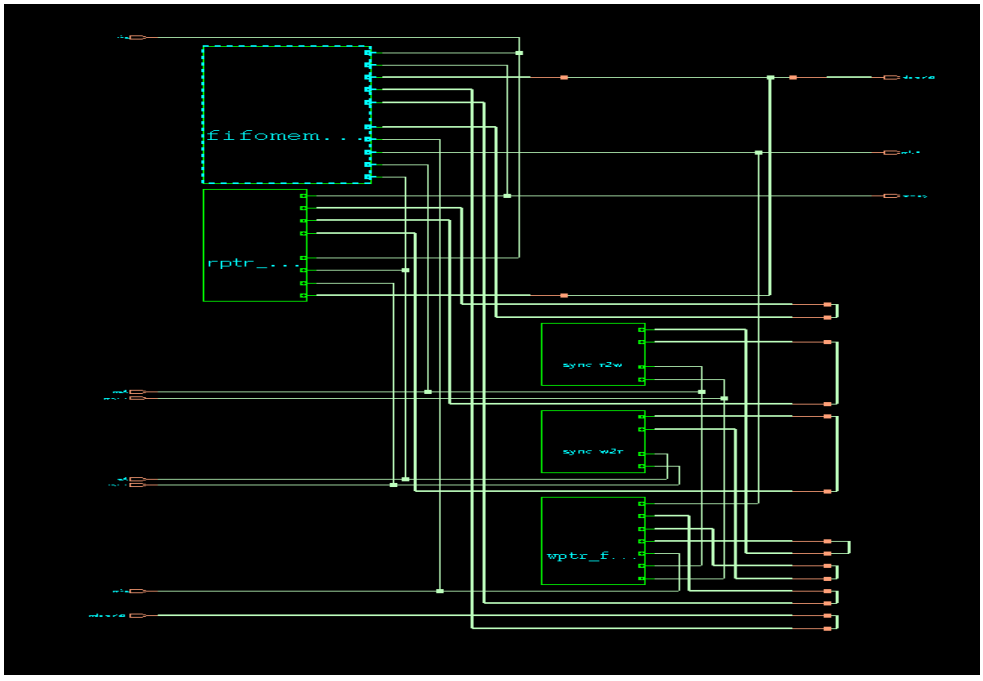

Figure 2. Asynchronous FIFO Synthesized Schematic

2. QOR Report:

a. Timing Analysis:

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Apr 27 2025  12:59:21 pm
  Module:                 async_fifo1
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Timing
--------

Clock  Period
--------------
clk   10000.0


  Cost    Critical         Violating
 Group   Path Slack  TNS     Paths
-------------------------------------
default    No paths   0.0
-------------------------------------
Total                 0.0          0
```

Figure 3. Asynchronous FIFO Timing Analysis Report

b. Area Analysis:

```
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Apr 27 2025  12:59:37 pm
  Module:                 async_fifo1
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


     Type      Instances  Area Area %
----------------------------------------
unresolved              5 0.000    0.0
physical_cells          0 0.000    0.0
----------------------------------------
total                   5 0.000    0.0
```

Figure 4. Asynchronous FIFO Area Analysis

c. Power Analysis:

```
Instance: /async_fifo1
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
     register     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
        logic     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
        clock     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 100.00%
  ------------------------------------------------------------------------
    Subtotal     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00 800.00%
  Percentage         100.00%      100.00%      100.00%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Figure 5. Asynchronous FIFO Power Analysis

Module 2: FIFO Memory

1. **Schematic after synthesis:**

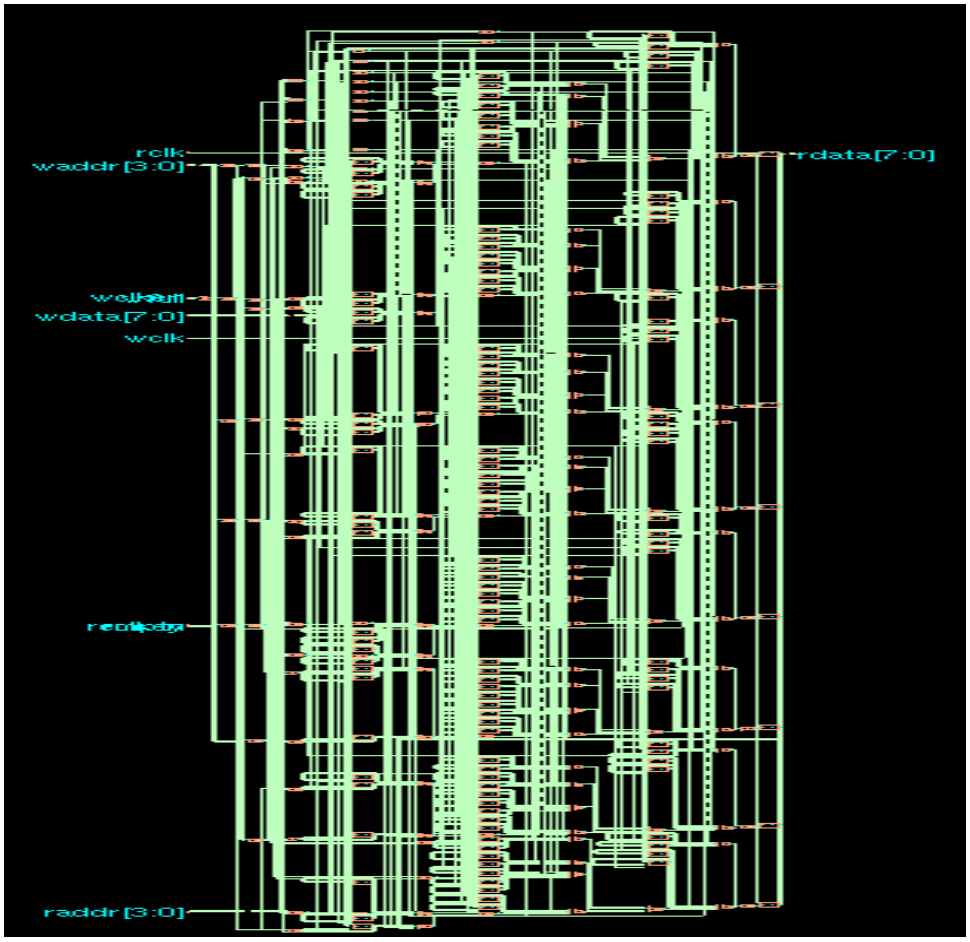

Figure 6. FIFO Memory Synthesised Schematic

2. **QOR Report:**
   a. **Timing Analysis:**

```
  Module:                 fifomem
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


Path 1: MET (9008 ps) Setup Check with Pin rdata_reg[3]/CK->D
     Startpoint: (R) mem_reg[2][3]/CK
          Clock: (R) clk
       Endpoint: (F) rdata_reg[3]/D
          Clock: (R) clk

                     Capture       Launch
       Clock Edge:+   10000            0
      Src Latency:+       0            0
      Net Latency:+       0 (I)        0 (I)
          Arrival:=   10000            0

            Setup:-    -116
     Required Time:=   10116
      Launch Clock:-       0
         Data Path:-    1108
             Slack:=    9008
```

Figure 7. FIFO Memory Timing Analysis

b. **Area Analysis:**

```
  Module:                 fifomem
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


  Gate    Instances    Area       Library
-----------------------------------------------
AND2X1            1     1.368    slow_vdd1v0
AOI221X1         16    38.304    slow_vdd1v0
AOI222X1          8    24.624    slow_vdd1v0
AOI22X1           8    16.416    slow_vdd1v0
AOI22XL          32    65.664    slow_vdd1v0
DFFQXL            8    43.776    slow_vdd1v0
INVXL             3     2.052    slow_vdd1v0
NAND2BX1          4     5.472    slow_vdd1v0
NAND2BXL          3     4.104    slow_vdd1v0
NAND2X1          14    14.364    slow_vdd1v0
NAND2XL          12    12.312    slow_vdd1v0
NAND3BXL          3     5.130    slow_vdd1v0
NAND4XL           8    13.680    slow_vdd1v0
NOR2BX1          12    16.416    slow_vdd1v0
NOR2X1            4     4.104    slow_vdd1v0
NOR2XL           12    12.312    slow_vdd1v0
NOR3XL            1     1.710    slow_vdd1v0
OR2X1             2     2.736    slow_vdd1v0
SDFFQX1         128   963.072    slow_vdd1v0
-----------------------------------------------
total           279  1247.616


     Type      Instances   Area   Area %
-----------------------------------------
sequential           136 1006.848   80.7
inverter               3    2.052    0.2
logic                140  238.716   19.1
physical_cells         0    0.000    0.0
-----------------------------------------
total                279 1247.616  100.0
```

Figure 8. FIFO Memory Area Analysis

c. **Power Analysis:**

```
Instance: /fifomem
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ------------------------------------------------------------------------------
      memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
    register     1.59689e-08  2.35051e-05  4.39641e-07  2.39607e-05  85.41%
       latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       logic     4.20993e-09  1.29217e-06  5.93772e-07  1.89016e-06   6.74%
        bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       clock     0.00000e+00  0.00000e+00  2.20320e-06  2.20320e-06   7.85%
         pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
          pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------------
    Subtotal     2.01789e-08  2.47973e-05  3.23661e-06  2.80541e-05 100.00%
  Percentage           0.07%       88.39%       11.54%      100.00% 100.00%
  ------------------------------------------------------------------------------
```

Figure 9. FIFO Memory Power Analysis

Module 3: Write Pointer Full Condition

1. **Schematic after synthesis:**

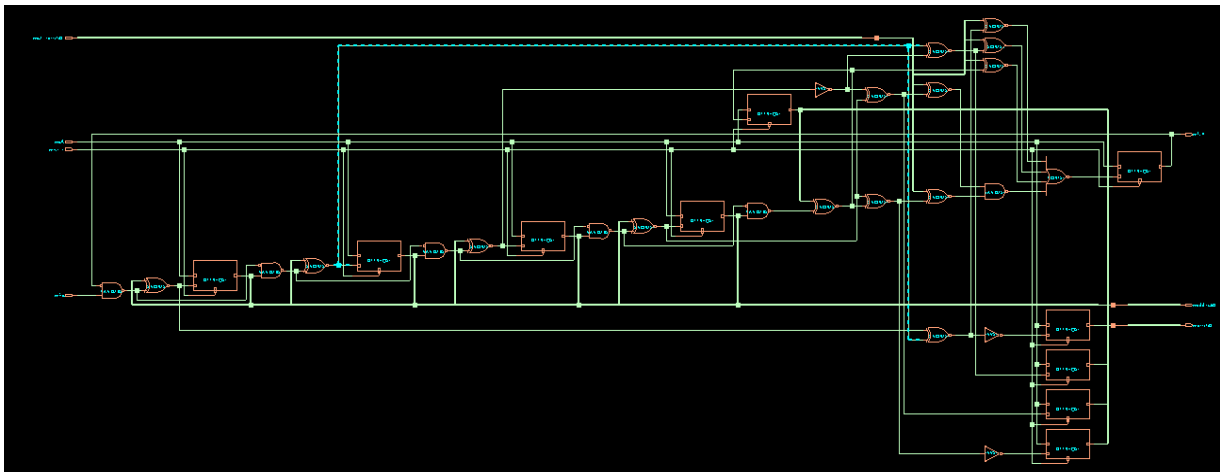

Figure 10. Write Pointer Full Condition Synthesized Schematic

2. **QOR Report:**
   a. **Timing Analysis:**

```
  Module:                   wptr_full
  Operating conditions:     PVT_0P9V_125C (balanced_tree)
  Wireload mode:            enclosed
  Area mode:                timing library
===============================================================


Path 1: MET (8375 ps) Setup Check with Pin wfull_reg/CK->D
     Startpoint: (R) wfull_reg/CK
           Clock: (R) clk
       Endpoint: (R) wfull_reg/D
           Clock: (R) clk

                      Capture       Launch
        Clock Edge:+    10000            0
       Src Latency:+        0            0
       Net Latency:+        0 (I)        0 (I)
           Arrival:=    10000            0

             Setup:-       48
     Required Time:=     9952
      Launch Clock:-        0
         Data Path:-     1576
             Slack:=     8375
```

Figure 11. Write Pointer Full Condition Timing Analysis Report

b. Area Analysis

```
  Module:                 wptr_full
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
==============================================================


  Gate    Instances   Area       Library
-----------------------------------------------
DFFRHQX1         10   61.560    slow_vdd1v0
INVX1             3    2.052    slow_vdd1v0
NAND2BX1          5    6.840    slow_vdd1v0
NAND2X1           1    1.026    slow_vdd1v0
NOR4X1            1    1.710    slow_vdd1v0
XNOR2X1          13   31.122    slow_vdd1v0
XOR2XL            1    2.736    slow_vdd1v0
-----------------------------------------------
total            34  107.046
```

Figure 12. Write Pointer Full Condition Area Analysis Report

c. Power Analysis:

```
Instance: /wptr_full
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
      memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
    register     1.33664e-09  3.10780e-06  2.50875e-08  3.13423e-06  77.56%
       latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       logic     1.14051e-09  5.64146e-07  1.79377e-07  7.44663e-07  18.43%
        bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       clock     0.00000e+00  0.00000e+00  1.62000e-07  1.62000e-07   4.01%
         pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
          pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------
    Subtotal     2.47715e-09  3.67195e-06  3.66464e-07  4.04089e-06 100.00%
  Percentage           0.06%       90.87%        9.07%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Figure 13. Write Pointer Full Condition Power Analysis Report

Module 3: Read Pointer Empty Condition

1. Schematic after synthesis:

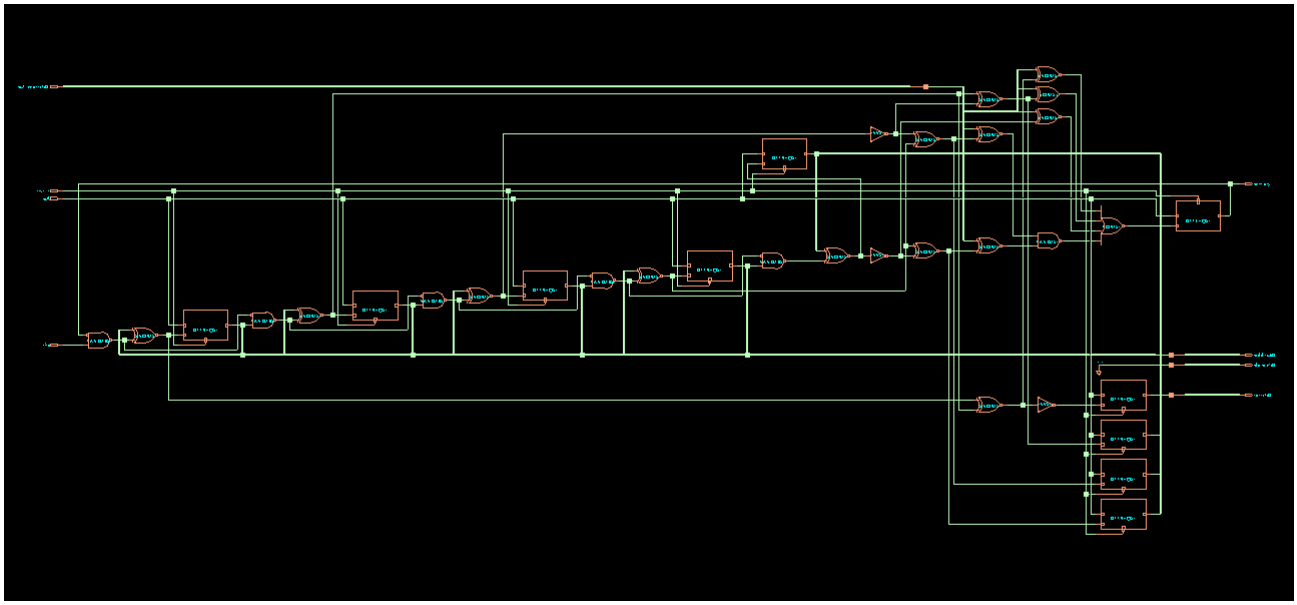

Figure 14. Read Pointer Empty Condition Synthesised Schematic

2. QOR Report

a. Timing Analysis

```
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Apr 27 2025  12:31:17 pm
  Module:                 rptr_empty
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
=============================================================


Path 1: MET (8331 ps) Setup Check with Pin rempty_reg/CK->D
     Startpoint: (R) rempty_reg/CK
          Clock: (R) clk
       Endpoint: (R) rempty_reg/D
          Clock: (R) clk

                     Capture       Launch
       Clock Edge:+    10000            0
      Src Latency:+        0            0
      Net Latency:+        0 (I)        0 (I)
          Arrival:=    10000            0

            Setup:-       44
    Required Time:=     9956
     Launch Clock:-        0
        Data Path:-     1625
            Slack:=     8331
```

Figure 15. Read Pointer Empty Condition Timing Analysis

b. Area Analysis

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Apr 27 2025  12:24:18 pm
  Module:                 wptr_full
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


  Gate    Instances   Area       Library
-------------------------------------------
DFFRHQX1         10   61.560    slow_vdd1v0
INVX1             3    2.052    slow_vdd1v0
NAND2BX1          5    6.840    slow_vdd1v0
NAND2X1           1    1.026    slow_vdd1v0
NOR4X1            1    1.710    slow_vdd1v0
XNOR2X1          13   31.122    slow_vdd1v0
XOR2XL            1    2.736    slow_vdd1v0
-------------------------------------------
total            34  107.046
```

Figure 16. Read Pointer Empty Condition Area Analysis Report

c. Power Analysis:

```
Instance: /rptr_empty
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
    register     1.35554e-09  3.06056e-06  2.50875e-08  3.08701e-06  77.36%
       latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       logic     1.14051e-09  5.62676e-07  1.77383e-07  7.41200e-07  18.58%
        bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       clock     0.00000e+00  0.00000e+00  1.62000e-07  1.62000e-07   4.06%
         pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
          pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ------------------------------------------------------------------------
    Subtotal     2.49604e-09  3.62324e-06  3.64471e-07  3.99021e-06 100.00%
  Percentage           0.06%       90.80%        9.13%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Figure 17. Read Pointer Empty Condition Power Analysis Report

Conclusion

- Successfully designed and implemented a parameterized asynchronous FIFO for safe multi-bit data transfer across asynchronous clock domains.
- Utilized Gray code pointers to ensure reliable pointer synchronization, avoiding metastability and ensuring accurate full/empty condition detection.
- Functional verification using Xilinx Vivado simulation confirmed correct FIFO operation under asynchronous conditions.
- Synthesis using Cadence Genus showed positive results in timing, area, and power analysis, validating the design's hardware feasibility.
- Modular design of FIFO (split into memory, pointer control, and synchronization modules) supports scalability and parameterization for different system requirements.

- Achieved a synthesizable, verifiable, and timing-closure friendly FIFO suitable for ASIC and FPGA real-world applications.
- Overall, the FIFO design meets industry standards for reliability, efficiency, and robustness in asynchronous data communication.