EE6306

APPLICATION SPECIFIC INTEGRATED CIRCUIT DESIGN

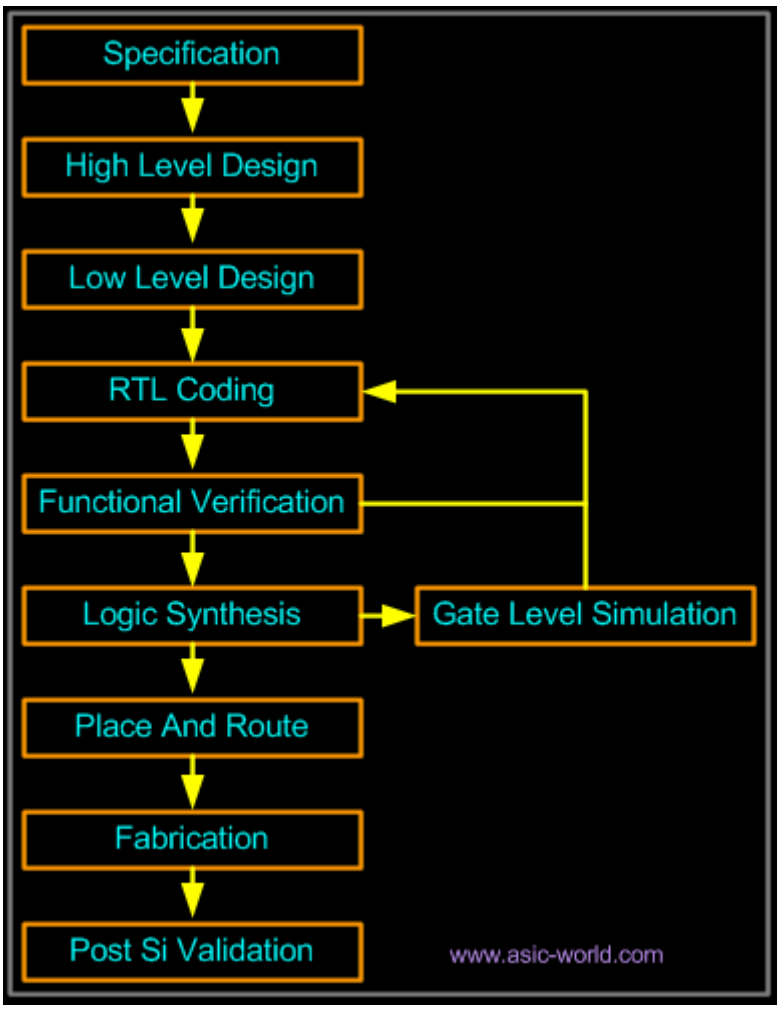
Design of a Mini Stereo Digital Audio Processor

PROJECT REPORT

TABLE OF CONTENTS

ASIC Design Flow

ASIC flow in designing an Integrated Circuit for the desired specification is summarized below:



Step 1: Specification

First step in the ASIC flow is to get the requirements of the application through analyzing. For IC design, getting the specification is a very significant step.

Step 2: High Level Design (Behavioral Description)

In this step in the ASIC design flow is to get design specifications based on the application being implemented. Behavioral modeling is a high-level modeling technique where behavior of the logic is modeled based on the specifications obtained. The behavioral logic is then tested by generating waveform and verifying the functionalities of the application. In this stage the structural implementation is not significant. In this step any redundant logics are removed.

Step 3: Low level design (RTL Design)

Register Transfer Logic (RTL) is used to describe the logic design part of the flow. Unlike behavioral modeling where code is written in high-level, RTL is written in transistor level. In this step in ASIC design flow the application specification is implemented in register level hence the name register – transfer level. Registers are responsible for storing values in it and all other computations are implemented in combinational logic.

Step 4: RTL verification

In this step the written RTL code is verified by checking its functionality. This can be achieved by writing test benches. Writing test benches must cover all the possible cases to determine its correct functioning of the application. If we find any flaw in the design, we should go back to the previous step and fix the RTL code. This is one of the significant step to implement a correct application. We can also generate timing, area and power details which will be useful in case of overhead issues.

Step 5: Logic synthesis

In this step, The RTL code is synthesized using the standard library and the physical realization of the RTL code can be seen. All the logics in the code will be mapped to the gates in the standard library. During synthesis with the help of logic optimization in the tool will help to speed up the logic and optimize the area. FSM decomposition, power and data-path optimization is done in this step. The tool will also makes use of area, timing and power information for efficient mapping to the gates and flip-flops.

Step 6: Floor planning Automatic Place and route

The gate level net-list which is obtained from the previous step is used for the physical implementation on the chip. Physical implementation includes partitioning, floor planning, placement and route. All these steps are using Synopsys IC compiler. Several iterations are run by the tool to reduce the overall wire length and area.

Route is first performed, this will give an abstract wire length and specified channel where actual routing should takes place. After this a detailed routing will takes place on the already routed lines. If there is more delay on this path different placement will be performed.

Design of Mini Stereo Digital Audio Processor

Objective

To design and implement an MSDAP in Hardware Descriptive Language by:

1. Defining the input & output ports and other specifications in detail.
2. Writing the behavioral model for MSDAP.
3. Writing a Test Bench to simulate all the cases of the specification.
4. Performing the Functional Verification to validate the specification.

Introduction

Most of the modern multimedia systems require the implementation of audio processors which are highly accurate and efficient being the first priority. In order to achieve this, the cost becomes invariantly high. So, when the same design is to be used for portable devices or household applications such as audio systems, cost of the audio processor and power consumption are of major concern. Most of the similar audio processing designs are implemented using the Digital Signal Processing (DSP) chips with separate left and right channel processing that performs the basic function of Finite Impulse Response (FIR) Digital Filter.

A new approach 'Mini Stereo Digital Audio Processor (MSDAP)' is proposed wherein a single MSDAP chip is capable of producing the same result as that of two DSP chips for two channels, thereby reducing the power and hardware and hence the cost of the chip. The FIR filter is implemented based on the following linear convolution:

$$y(n) = \sum_{k=0}^{N} h(k) \times x(n-k)$$

Where,
x (n) = input audio samples
y (n) = output audio samples
h (k) = Filter co-efficients
N = Filter order

It follows from the above expression that N additions and (N +1) multiplications are required. In hardware, implementing the multiplication either floating or fixed point is very costly with regard to power, area and complexity. The proposed model can overcome these issues by replacing the multiplication unit with a one bit shifter and either an addition/subtraction unit which proves to be much faster and efficient than the former.

Design Specifications

Our application of the hearing aid has the MSDAP chip as the critical part. The system specification for this application can be divided into three parts –

1. System level view and algorithm
2. Operation modes and FSM
3. Pins, data formats and signals

System level view and algorithm

Algorithm Description:

The MSDAP will focus on the FIR Digital filtering for the hearing aid application, which involves the following linear convolution:

$$y(n) = \sum_{k=0}^{N} h(k) \times x(n-k)$$

Where $x(n)$ and $y(n)$ are the input and output audio sequences, and $h(k)$ are filter coefficients with the filter order N.

The steps involved in computing the linear convolution shown above is discussed in detail below:

$$y(n) = h(0)x(n-0) + h(1)x(n-1) + h(2)x(n-2) + \ldots\ldots + h(\mathrm{N})x(n-\mathrm{N})$$

$h(0)$, $h(1)$, $h(2)\ldots$ $h(\mathrm{N})$ are expressed in terms of POT.

In order to reduce the hardware complexity of using a shifter of up to 16 bits, a one bit shifter is employed in the design. Accordingly, the above expression is transformed to –

$$y(n) = 2^{-1}(\ldots.2^{-1}(2^{-1}(2^{-1}u_1 + u_2) + u_3) + \ldots.) + u_{16})$$

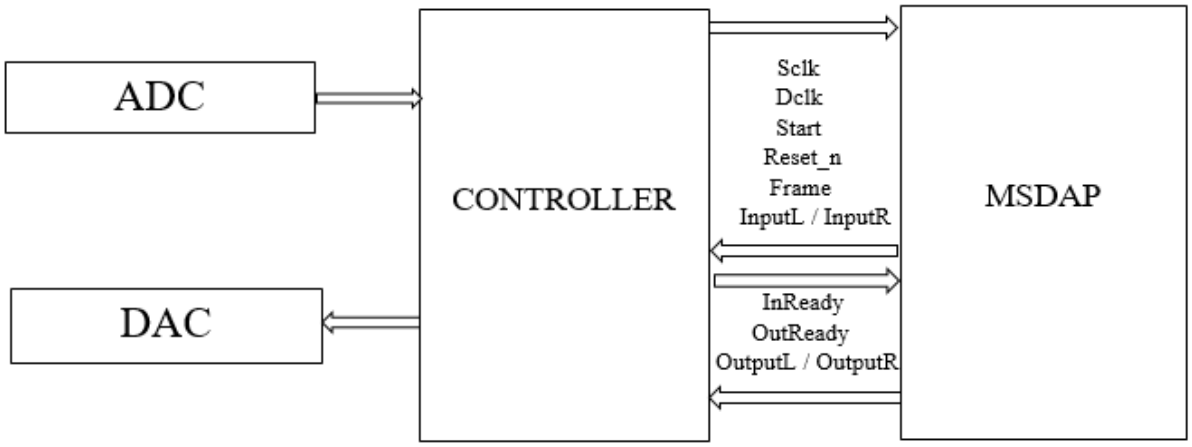
Where $u_0, u_1, u_2, u_3 \ldots u_{16}$ are addition/subtraction of the input samples defined by the r_j values.

$$u_j = x_j(1) + x_j(2) + \ldots. + x_j(r_j) \qquad 1 \leq j \leq 16$$

Where $x_j(l) \in \{\pm x(n-k)\}$, $1 \leq l \leq r_j$

With this approach, MSDAP can implement two FIR filters for left and right channels and the order of the filter can go up N = 255.

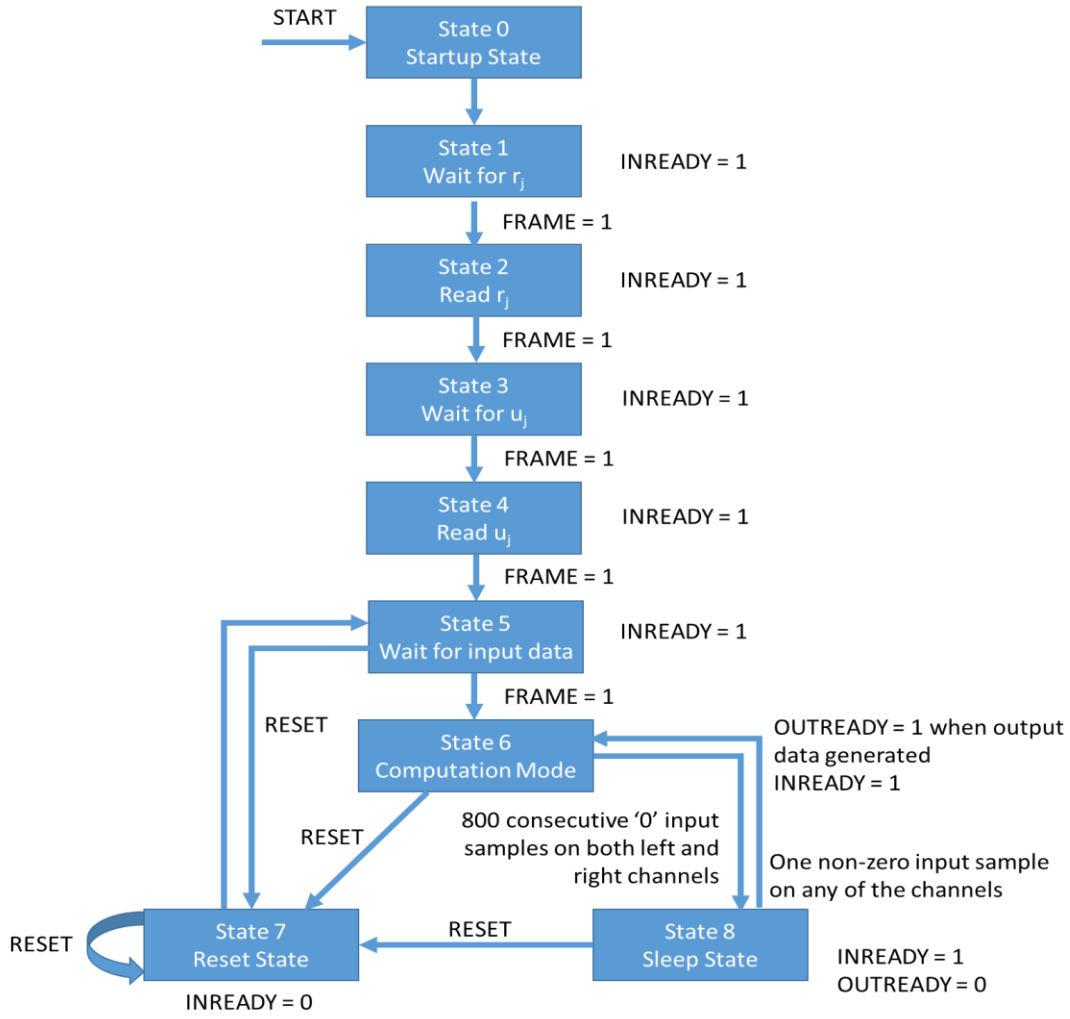
Block Diagram:



As we can see from the block diagram above, we have input signals Sclk, Dclk, Start, Reset_n, Frame, InputL/R and output signals Inready, Outready, OutputL/R to and from the MSDAP chip respectively, from the controller system.

Operation modes and FSM

State Description of FSM:



Operation modes:

1. **State 0 (Startup State):**
   The START signal, when set high, initializes the chip and prepares it for working by clearing memories and registers, and then the chip enters State 1.

2. **State 1 (Wait for r_j):**
   INREADY is set high. If FRAME = 1 is seen on the line, the chip enters state 2.

3. **State 2 (Read r_j):**
   The chip starts reading r_j values and INREADY remains high. Once all the r_j values have been collected, the chip enters state 3.

4. **State 3 (Wait for u_j):**
   INREADY is set high. If FRAME = 1 is seen on the line, the chip enters state 4.

5. **State 4 (Read u_j):**
   The chip starts reading u_j values and INREADY remains high. Once all the u_j coefficient values have been collected, the chip enters state 5.

6. **State 5 (Wait for input data):**
   INREADY is set high. If FRAME = 1 is seen on the line, the chip enters State 6. Or else, if RESET_N = 0 is seen, the chip enters State 7.

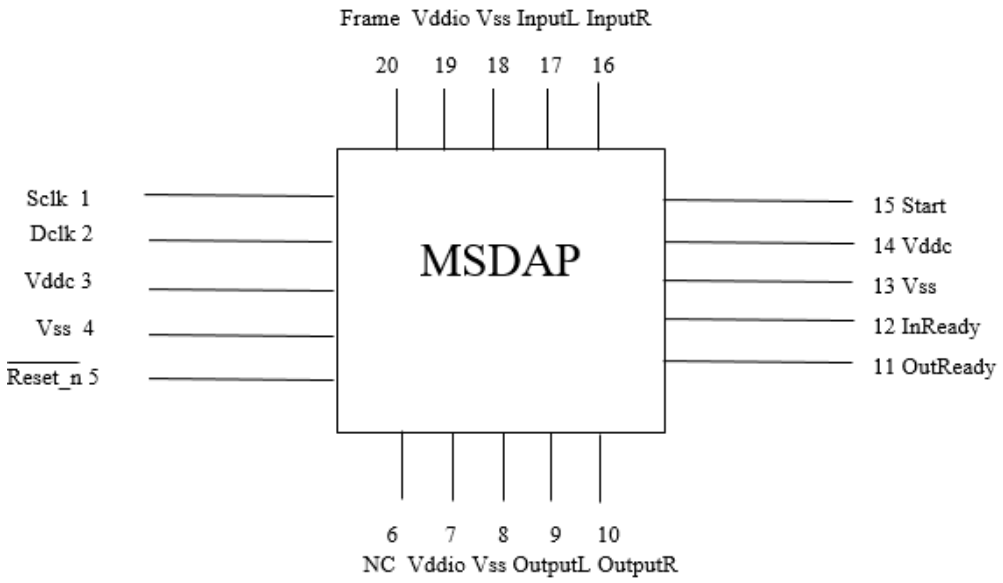
7. **State 6 (Computation mode):**
   Here the chip accepts incoming input data, computes the output values for convolution, and puts out output values too. INREADY remains high. If RESET_N = 0 is detected, the chip enters State 7. Or else, if 800 consecutive input samples on both channels are detected as zero, the chip enters the Sleep mode (State 8).

8. **State 7 (Reset mode):**
   Here, INREADY is set low. All registers and memories except r_j and coefficients are cleared. If RESET_N = 0 is seen on the line, the chip restarts the same process. Once reset is completed, the chip automatically goes to State 5.

9. **State 8 (Sleep mode):**
   Here, the chip in sleep mode until a non-zero input sample is detected on either left or right channel. INREADY is kept high. If RESET_N = 0 is detected, the chip goes to State 7.

Pin assignment and description

Pin Diagram:



Function of Signals:

- Sclk - 1: Sclk is the system clock that is used to synchronize all the control signals and IO signals. All the state changes and the computation happen with the negative or falling edge of the System clock. The frequency of the Sclk determines the frequency of operation of the entire system/ chip. This is an input signal to the MSDAP from the controller.

- Dclk - 2: Dclk is the Data clock that is used as the timing reference to receive and send the input samples. A bit is received or sent at the falling edge of the Dclk. Hence to receive a 16 bit data, 16 Dclk cycles are required. One Dclk is 35 Sclk cycles. This is an input signal to the MSDAP from the controller.

- Frame - 20: Frame is an input signal from the controller to the MSDAP. While sending or receiving the input or output to or from the MSDAP, the Frame is set high. Frame signifies the start of the first bit of data-input, output, Rj or co-efficient. Frame is used as active high signal and is made high for one Dclk cycle. After receiving the first bit of the data, the Frame goes inactive i.e. low. Frame is made active after every 16 Dclk cycles and hence a 16 bit data is received between two successive Frames.

- Start - 15: Start is an asynchronous input signal from the Controller to MSDAP. It is used to indicate the start of the FSM and initialization of chip begins. Until then the chip is said to be idle. Start is an active high input signal.

- $\overline{\text{Reset\_n}}$ - 5: Reset is an active low asynchronous input signal to the MSDAP. The chip goes to the reset mode when this signal is applied.

- InputL - 17: This is a one bit wide left input channel. It allows left co-efficient, Rj values and left input audio samples in serial format.

- InputR - 16: This is same as InputL but used for the right audio channel.

- InReady - 12: InReady is an input signal to the Controller from the MSDAP which is set high when the chip is ready to receive the data from both the channels. InReady is aligned with the Frame.

- OutputL - 9: This is a one bit wide left output channel. It allows left output audio samples in serial format.

- OutputR - 10: This is same as OutputL but used for the right audio channel.

- OutReady - 11: It is an output signal from the MSDAP to the Controller that goes high when the output is ready to be sent. It goes low when the last bit of the output data sample is sent. This signal is aligned with the rising edge of the Frame signal.

Signal and Data Format

All the input data x (n), coefficients u_j, and r_j values are transmitted as 16-bit 2's complement form values. Zero-padding is done to bring each value to 16 bits. The values are received by the chip with the MSB leading in first. All the input received must be synchronized to the data clock (Dclk), which is given as 768 kHz per bit. Since there are 16 bits for each value, the frame is 16 bits long, and the frame rate will be 768/16 = 48 kHz. However, the Sclk will depend on how the system is implemented.

1. **R_j format** –
   The r_j value is 8 bits long and is stored in the lower half of the 16-bit frame, while the other higher order bits are zero padded to form the frame. The frame is as shown below –

0 7	8 15
Unused/Padded bits	MSB R_j data LSB

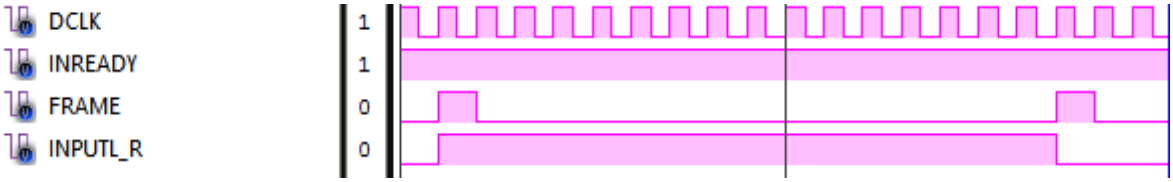
2. **U_j format** –
   The coefficient u_j value is 8 bits long with a sign bit, and the rest of the frame is zero padded. The 16-bit frame is as shown below –

0 6	7	8 15
Unused/Padded bits	Sign bit	MSB Coefficient data LSB

3. **Input/Output data format** –
   The input/output data frames are 16 bit 2's complement values.

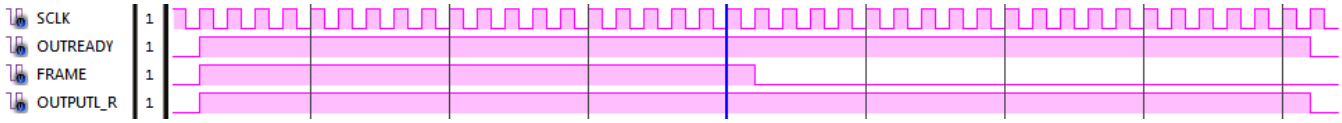
0 15
MSB Input/output data bits LSB

The timing diagram for the input is as given below –



As we see, the FRAME signal is raised high for one clock cycle whenever a new frame with new data starts. The data is transmitted as one bit per clock cycle for 16 cycles of DCLK i.e. for 16 bits. During this time, the INREADY signal is kept high to signal that inputs are ready to be accepted.

The timing diagram for the output is as follows –



As we see, the FRAME signal works according to DCLK (16 cycles), while OUTREADY and OUTPUTL/R are synchronized according to the system clock SCLK. SCLK depends on the architecture of the implementation. Once OUTREADY is set high, OUTPUTL/R gives the 40-bit output, and once all 40 bits are put on the line, one every clock cycle, and then OUTREADY is set low.

System Setting

The main blocks in the system is ADC, Controller, MSDAP and DAC (Refer Block Diagram above)

ADC: Analog to Digital Converter is used to convert the analog audio samples to digital values. The analog data is sampled at the rate of 48 KHz.

Controller: Controller is responsible for generating the control signals to the MSDAP. It also sends the converted digital samples from ADC to the MSDAP at the rate of Dclk for the computation. After the computation, MSDAP sends the output samples to the controller at the rate of Dclk which is 768 KHz.

MSDAP: MSDAP is responsible for the computation on the received input samples. The computation is carried out at the rate of Sclk which is 26.88MHz. For one complete computation on the input data sample, the number of Sclk cycles required will determine the frequency of operation of MSDAP chip. For one computation, 512 additions and 16 shifts are required. Rj value add upto a total of 512, requiring 512 additions/subtractions to calculate 16 U values. After each U value calculation, it is to be shifted once. Hence, a total of 528 Sclk cycles are required. Based on these calculations, we decided the Sclk frequency to be 26.88MHz and Dclk frequency to be 768 KHz.

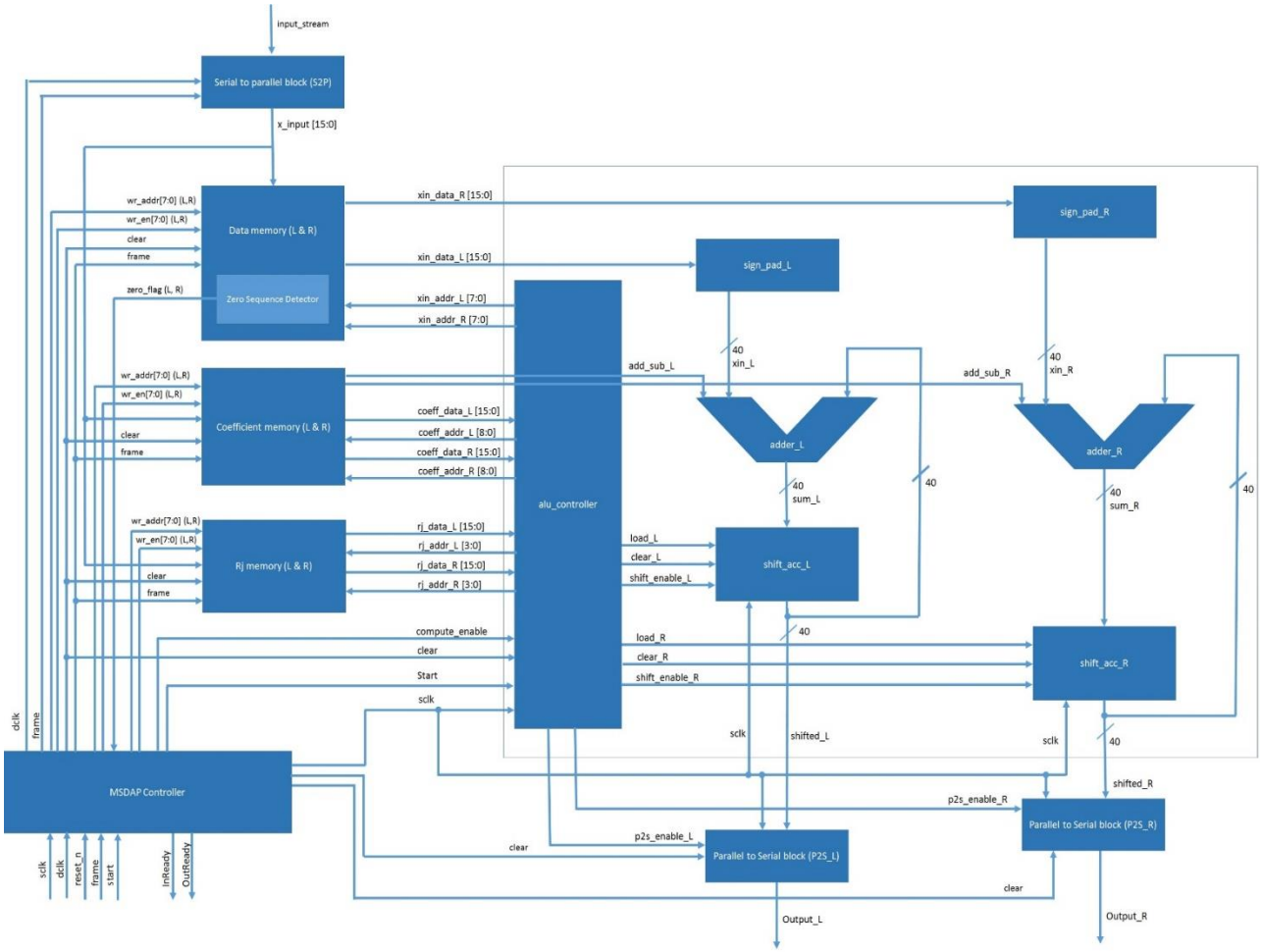
DAC: Once the output digital samples are received by the Controller, it is fed to the Digital to Analog Converter that converts the digital samples to actual audio output signals.

Architecture of MSDAP

The architecture for MSDAP contains the following blocks:

- MSDAP Control Unit
- Memory Bank
- Serial In Parallel Out (SIPO) Unit
- Zero Detector Unit
- ALU Control Unit
- Sign Extend Unit
- Adder Unit
- Accumulate and Shift Unit
- Parallel In Serial Out (PISO) Unit.

Below figure shows the architecture of MSDAP for both left and right channels.



Definition and Specification of each functional block:

1. **Serial In Parallel Out [S2P]:**

Functionality:

This functional block operates on the rising edge of the Dclk and the FRAME. Serial data starting from the MSB of the 16bits input data is received at the arrival of frame and it is captured in the rising edge of the Dclk. Once the frame goes high, the MSB data is received on that edge and the rest 15 bits of data is received at every Dclk positive edge. Once the data is packed into a 16 bit word, the S2P block signals the FSM Controller to enable the respective data memory and passes the address.

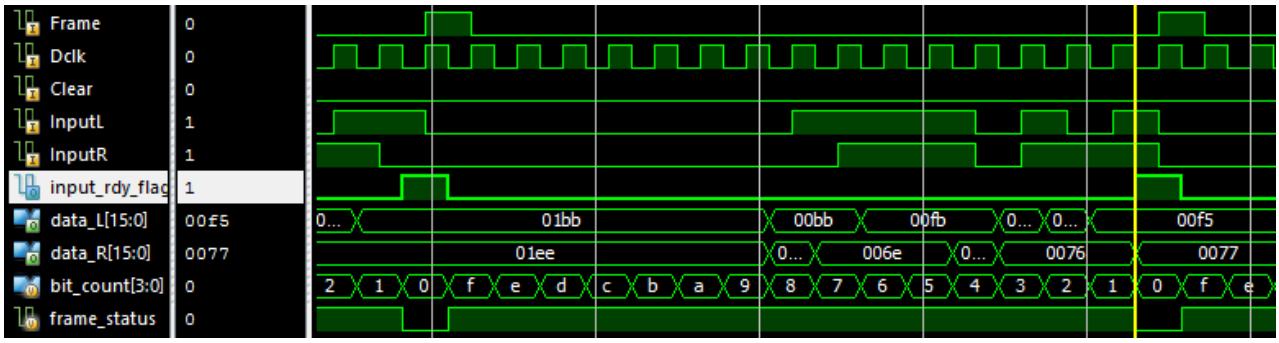
Signal	Input/output	Description
Frame	Input	Frame is aligned with the Dclk and the input data. For the first bit from 16 bit word data, the Frame will be active, after first Dclk clock cycle, the Frame goes inactive.
Clear	Input	Clear is used to clear the SIPO whenever a reset condition arises from the main controller
Dclk	Input	Dclk is the timing reference for receiving the input data serially. Data is received serially at the positive edge of the Dclk.
InputL, InputR	Input	InputL, InputR are the serial inputs to the S2P. These input data are captured on the positive edge of Dclk. MSB at the 0^{th} position is transmitted first and the LSB at 15^{th} position is transmitted last.
data_L, data_R [15:0]	Output	data_L, data_R are the 16 bit parallel data fed as data to be written to the memory.
input_rdy_flag	Output	input_rdy_flag is used to signal to the main controller that data is ready to write to memory

Verilog Code:

```
module SIPO (Frame, Dclk, Clear, InputL, InputR, data_L, data_R, input_rdy_flag);
	input Frame, Dclk, Clear, InputL, InputR;
	output reg input_rdy_flag;
	output reg [15:0] data_L;
	output reg [15:0] data_R;
	reg [3:0] bit_count;
	reg frame_status;

	always @(negedge Dclk or posedge Clear)
	begin
		if (Clear == 1'b1)
		begin
			bit_count = 4'd15;
			//temp_L = 16'd0;
			//temp_R = 16'd0;
			data_L = 16'd0;
			data_R = 16'd0;
			input_rdy_flag = 1'b0;
			frame_status = 1'b0;
		end
		else
		begin
			if (Frame == 1'b1)
```

```
                        begin
                                bit_count = 4'd15;
                                input_rdy_flag = 1'b0;
                                data_L [bit_count] = InputL;
                                data_R [bit_count] = InputR;
                                frame_status = 1'b1;
                        end
                        else if (frame_status == 1'b1)
                        begin
                                bit_count = bit_count - 1'b1;
                                data_L [bit_count] = InputL;
                                data_R [bit_count] = InputR;
                                if (bit_count == 4'd0)
                                begin
                                        //data_L = temp_L;
                                        //data_R = temp_R;
                                        input_rdy_flag = 1'b1;
                                        frame_status = 1'b0;
                                end
                                else
                                begin
                                        //data_L = data_L;
                                        //data_R = data_R;
                                        input_rdy_flag = 1'b0;
                                        frame_status = 1'b1;
                                end
                        end
                        else
                        begin
                                bit_count = 4'd15;
                                data_L = 16'd0;
                                data_R = 16'd0;
                                input_rdy_flag = 1'b0;
                                frame_status = 1'b0;
                        end
                end
        end
endmodule
```

Simulation Results:



Operation starts at the beginning of Frame appearing and is completed when input_rdy_flag is raised and data_L and data_R is considered valid for writing to memory.

2. Rj Memory (L & R):

Functionality:

Rj block is 16 bits wide and a depth of 16 locations. It is used to save the Rj values. There are 2 such blocks for left and right channel Rj samples. This Block is insensitive to the clear signal from the FSM controller. Based on the address and write enable, the data is being written into the memory at the positive edge of the Sclk. Data is read out for the computation by using read address signal. 4 bits address is required.

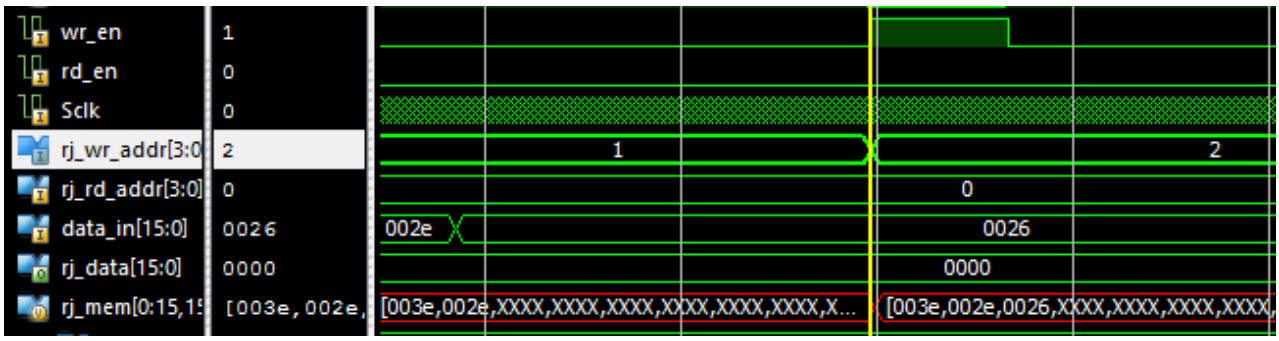
Signal	Input/output	Description
Sclk	Input	Sclk is the timing reference for the rj memory.
data_in [15:0]	Input	data_in is a 16 bit input word from the S2P. Based on the address and the write enable signals, the x_input is written to either rj memory at the positive edge of the Sclk.
rd_en	Input	rd_en is used by the ALU to enable reading from the memory
rj_wr_addr [3:0] (L/R)	Input	Address is 4 bits wide. This address comes from the MSDAP controller. It is used to write the data into the block. There are two signals for both right and left channel.
wr_en (L/R)	Input	Write enable is used by the main controller to enable writing to the memory
rj_data [15:0] (L/R)	Output	rj_data_L, rj_data_R is the 16 bit data from the left and right Rj memory respectively and is fed to the left and right ALU Control Block. The data is read when the Frame goes high.
rj_rd_addr [3:0] (L/R)	Input	rj_rd_addr is 4 bit address from the ALU controller. This is used to fetch the corresponding data from the left and right rj memory.

input_rdy_flag	Output	input_rdy_flag signals the main controller that the input is ready for processing

Verilog Code:

```
module rj_memory (input wr_en, rd_en, Sclk,
                                        input [3:0] rj_wr_addr, rj_rd_addr,
                                        input [15:0] data_in,
                                        output [15:0] rj_data);

        reg [15:0] rj_mem [0:15];

        always @(negedge Sclk)
        begin
                if(wr_en == 1'b1)
                        rj_mem[rj_wr_addr] = data_in;
                else
                        rj_mem[rj_wr_addr] = rj_mem[rj_wr_addr];
        end

        assign rj_data = (rd_en) ? rj_mem[rj_rd_addr] : 16'd0;
endmodule
```

Simulation Results:



As we see, when wr_en is enabled, the rj_wr_addr is used to write the data 0026 to the rj_mem.

3. Co-efficient Memory (L & R):

Functionality:

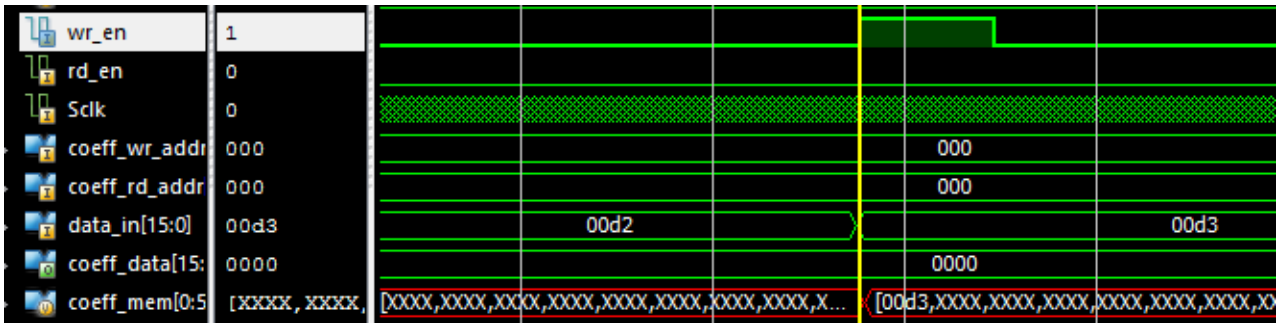
Co-efficient Block is 16 bits wide and a depth of 512 locations. Based on the address and write enable, the data is being written into the memory at the positive edge of the Sclk. Data is read out for the computation by using read address signal. There are 2 such blocks for left and right channel Co-efficient values. 9 bits address is required.

Signal	Input/output	Description
Sclk	Input	Sclk is the timing reference for the coeff memory.
data_in [15:0]	Input	data_in is a 16 bit input word from the S2P. Based on the address and the write enable signals, the data_in is written to either coeff memory at the positive edge of the Sclk.
rd_en	Input	rd_en is used by the ALU to enable reading from the memory
coeff_wr_addr [8:0] (L/R)	Input	Address is 9 bits wide. This address comes from the MSDAP controller. It is used to write the data into the block. There are two signals for both right and left channel.
wr_en (L/R)	Input	Write enable is used by the main controller to enable writing to the memory
coeff_data [15:0] (L/R)	Output	coeff_data_L, coeff_data_R is the 16 bit data from the left and right coeff memory respectively and is fed to the left and right ALU Control Block. The data is read when the Frame goes high.
coeff_rd_addr [8:0] (L/R)	Input	coeff_rd_addr is 9 bit address from the ALU controller. This is used to fetch the corresponding data from the left and right coeff memory.
input_rdy_flag	Output	input_rdy_flag signals the main controller that the input is ready for processing

Verilog Code:

```
module coeff_memory (input wr_en, rd_en, Sclk,
                                                input [8:0] coeff_wr_addr, coeff_rd_addr,
                                                input [15:0] data_in,
                                                output [15:0] coeff_data);

       reg [15:0] coeff_mem [0:511];

       always @(negedge Sclk)
       begin
              if(wr_en == 1'b1)
                     coeff_mem[coeff_wr_addr] = data_in;
              else
                     coeff_mem[coeff_wr_addr] = coeff_mem[coeff_wr_addr];
       end

       assign coeff_data = (rd_en) ? coeff_mem[coeff_rd_addr] : 16'd0;
endmodule
```

Simulation Results:



As we see, when wr_en is enabled, the coeff_wr_addr is used to write the data 00d3 to the coeff_mem at location 000.

4. **Data Memory (L & R):**

Functionality:

Data block is 16 bits wide and a depth of 256 locations. Each time a Frame is received by the SIPO from the FSM controller, upon activation of write enable signal and address, the data is being written to the Data block at the positive edge of the Sclk. Once the Data block is filled, it is re-used for the new incoming data from the 0th location. Hence, it acts like a circular / cyclic buffer. There 2 such data blocks for left and right channel input data samples. When a clear signal is sent by the FSM controller, all the data in the Data Block memory is cleared. 8 bits address is required.

Signal	Input/output	Description
Sclk	Input	Sclk is the timing reference for the data memory.

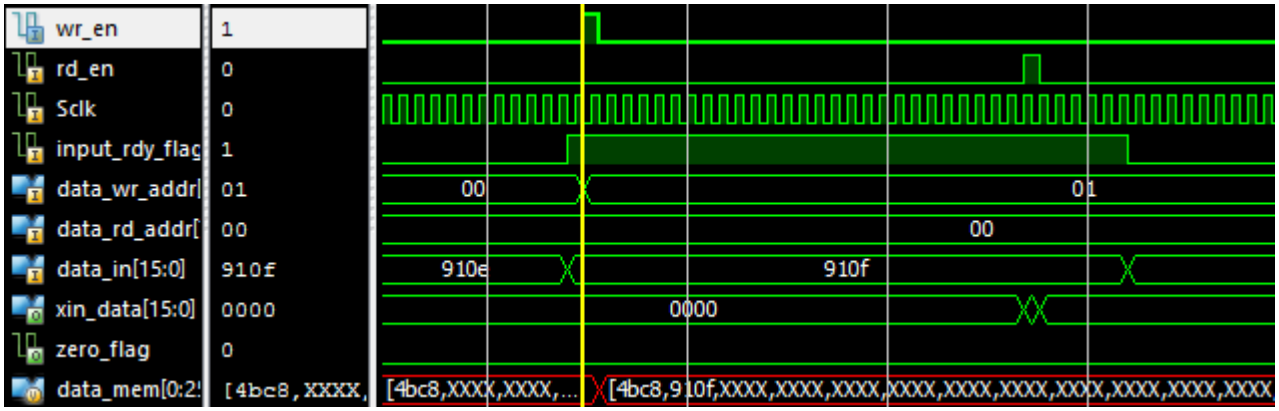
data_in [15:0]	Input	data_in is a 16 bit input word from the S2P. Based on the address and the write enable signals, the data_in is written to either data memory at the positive edge of the Sclk.
rd_en	Input	rd_en is used by the ALU to enable reading from the memory
data_wr_addr [7:0] (L/R)	Input	Address is 8 bits wide. This address comes from the MSDAP controller. It is used to write the data into the block. There are two signals for both right and left channel.
wr_en (L/R)	Input	Write enable is used by the main controller to enable writing to the memory
xin_data [15:0] (L/R)	Output	xin_data_L, xin_data_R is the 16 bit data from the left and right data memory respectively and is fed to the left and right ALU Control Block. The data is read when the Frame goes high.
data_rd_addr [7:0] (L/R)	Input	coeff_rd_addr is 8 bit address from the ALU controller. This is used to fetch the corresponding data from the left and right data memory.
input_rdy_flag	Output	input_rdy_flag signals the main controller that the input is ready for processing
zero_flag	Output	zero_flag is used to enforce the 800 zeroes condition for sleep mode

Verilog Code:

```
module data_memory (input wr_en, rd_en, Sclk, input_rdy_flag,
                                 input [7:0] data_wr_addr, data_rd_addr,
                                 input [15:0] data_in,
                                 output [15:0] xin_data,
                                 output reg zero_flag);

       reg [15:0] data_mem [0:255];
       reg [11:0] zero_cnt;

       always @(negedge Sclk)
       begin
```

```
            if(wr_en == 1'b1)
                  data_mem[data_wr_addr] = data_in;
            else
                  data_mem[data_wr_addr] = data_mem[data_wr_addr];
      end

      always @(posedge input_rdy_flag)
      begin
            if (data_in == 16'd0)
            begin
                  zero_cnt = zero_cnt + 1'b1;
                  if (zero_cnt == 12'd800)
                        zero_flag = 1'b1;
                  else if (zero_cnt > 12'd800)
                  begin
                        zero_cnt = 12'd800;
                        zero_flag = 1'b1;
                  end
            end
            else if (data_in != 16'd0)
            begin
                  zero_cnt = 12'd0;
                  zero_flag = 1'b0;
            end
      end

      assign xin_data = (rd_en) ? data_mem[data_rd_addr] : 16'd0;
endmodule
```

Simulation Results:



As we see, when wr_en is enabled, the data_wr_addr is used to write the data 910F to the data_mem at location 01. Also, rd_en is enabled to read the data from location 00 i.e. 4BC8.

5. Adder Block:

Functionality:

Addition/Subtraction is performed in this block depending on the add_sub_L or add_sub_R signal from the coefficient memory block. The total number of additions or subtractions in this unit is 512 for one computation of the input data sample. Addition/subtraction takes place at the positive edge of the Sclk. One of the operands come from the sign extension block and the other operand comes from the Shifter block. Both operands are 40 bits wide. The result which is 40 bits is fed back to the Shifter block.

Adder (L & R)

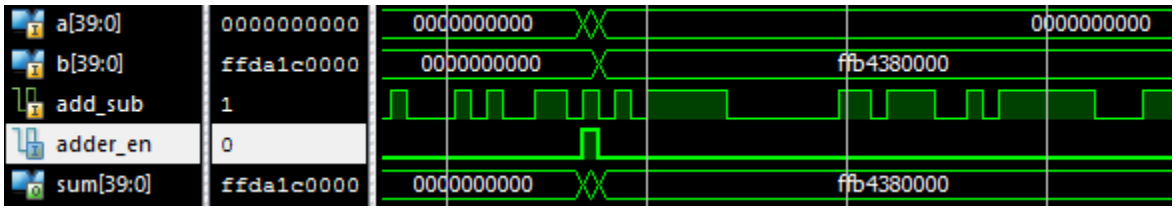
Signal	Input/Output	Description
add_sub	Input	add_sub_L is 1 bit wide coming from the ALU controller based on the MSB of the coefficient value read from memory. This bit decides whether addition or subtraction that has to be performed between the operands of the adder_L.
a	Input	a is 40 bits wide which is sign extended and padded version of xin_data_L coming from the sign_pad_L block. This is one of the operands for adder_L.
b	Input	b is 40 bits wide which is fed as another operand to the adder_L block.
adder_en	Input	adder_en is used to enable the adder to perform addition/subtraction at required times
sum	Output	This is the result of a and b which is 40 bits wide. This data is fed to the shift_acc_L block.

Verilog Code:

```
module adder(
       input [39:0] a,
    input [39:0] b,
    input add_sub,
       input adder_en,
    output [39:0] sum
    );

              assign sum = (add_sub == 1'b1) ? (b - a) :
```

```
                (add_sub == 1'b0) ? (b + a) :
                sum;

endmodule
```

Simulation Results:



The addition operation is carried out when adder_en is activated. As we can see, subtraction is carried out i.e. (b – a) to retain the value 0xFFB4380000.

6. Shift and Accumulate block:

Functionality:

The content in this block is initially set to 40 bit zero. It holds the result of the adder block when a load signal from the ALU control block is applied to it. One bit is right shifted whenever shift_enable signal from the ALU controller is applied. Shifting is done in the positive edge of the Sclk. Whenever there is a Reset signal from the ALU control block, the contents in the accumulator is cleared. The shiftout_L or shiftout_R is 40 bits which is fed back as one of the operand to the respective adder block.

shift_acc (L & R)

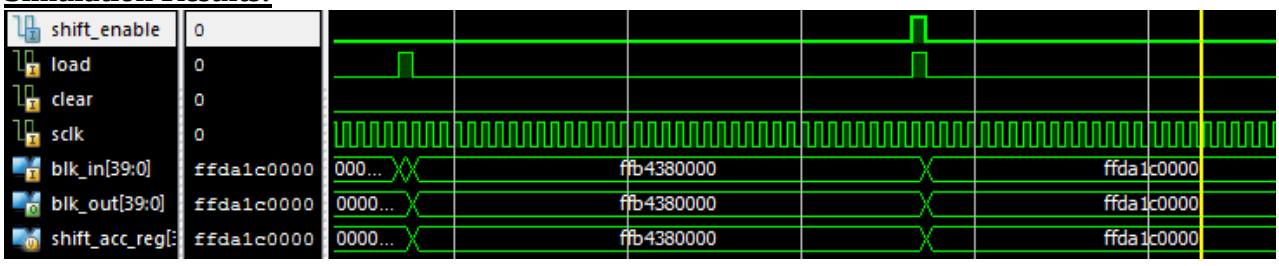
Signal	Input/output	Description
shift_enable	Input	shift_enable signal indicates that shift operation has to be done in the shift_acc block.
load	Input	load signal indicates that accumulator register is loaded with the result of the adder block.
clear	Input	clear signal indicates that accumulator register which is of 40 bits wide has to be made zero. It is also used when global reset is applied.

sclk	Input	Sclk is the timing reference signal to perform right shift operation on the data in the 40 bit accumulator register.
sum	Input	This is the result of xin and shifted signal which is 40 bits wide. This data is fed to the shift_acc block.
shifted	Output	This is 40 bits wide which is sent to the PISO after the computation is done. The 40 bits is also fed back as one of the operand to adder block.

Verilog Code:

```
module shift_acc(
    input shift_enable,
    input load,
    input clear,
    input sclk,
    input [39:0] blk_in,
    output [39:0] blk_out
    );

       reg [39:0] shift_acc_reg;

       always @(posedge sclk)
       begin
              if (clear)
                     shift_acc_reg = 40'd0;
              if (load && shift_enable)
                     shift_acc_reg = {blk_in[39], blk_in[39:1]};
              else if (load && !shift_enable)
                     shift_acc_reg = blk_in;
              else
                     shift_acc_reg = shift_acc_reg;
       end

       assign blk_out = shift_acc_reg;

endmodule
```

Simulation Results:



We see that as shift_enable is asserted, the shift is carried out.

7. ALU Controller:

Functionality:

This block receives the required signals from FSM control block and the data from the memory block. The ALU control block is activated by the FSM control block whenever the computation on the input data sample has to be started.

alu_controller

Signal	Input/output	Description
compute_enable	Input	This is a 1 bit enable signal from FSM control block to activate the ALU control block. This signal will be high whenever computation is started on the input data sample.
Clear	Input	Clear signal indicates that ALU control block has to send clear_L and clear_R signal to the accumulator to clear the data in the accumulator.
rj_data_L, rj_en_L rj_data_R, rj_en_R	Input	rj_data_L and rj_data_R are the 16 bit data from the left rj block and right rj block respectively and is fed to the alu_controller block, while rj_en are enables to the memories
coeff_data_L, coeff_en_L coeff_data_R, coeff_en_R	Input	This is a 16 bit data from the left coefficient block and right coefficient block and are fed to the ALU control block based on the rj values, while coeff_en are the enables to the memories
sclk	Input	sclk is the timing reference signal to the ALU control block to perform load or shift operation.
sleep_flag	Input	sleep_flag indicates to the ALU that the main controller is now in sleep mode and all operations must stop
rj_addr_L, rj_addr_R	Output	rj_addr_L and rj_addr_R are the 4-bit addresses to the left rj block and right rj block respectively

coeff_addr_L, coeff_addr_R	Output	These are 9 bit addresses to the left coefficient block and right coefficient block based on the rj values.
xin_addr_L, xin_en_L, xin_addr_R, xin_en_R	Output	This is an 8 bit read address to access input data with respect to left and right channels. Once the address is passed to the memory block, the data from the memory can be accessed. xin_en are the enables to the memories
add_inp_L, add_inp_R	Output	These signals represent the sign extended and padded forms of the input data coming in, and are given to the adder blocks respectively.
shift_enable_L, shift_enable_R	Output	shift_enable_L and shift_enable_R signals indicate that shift operation has to be done in the shift_acc_L or shift_acc_R block.
load_L, load_R	Output	load_L or load_R signal indicates that accumulator register is loaded with zero. This signal is aligned with Frame.
clear_L, clear_R	Output	clear_L or clear_R signal indicates that accumulator register which is of 40 bits wide has to be made zero.
p2s_enable_L, p2s_enable_R	Output	This is 1 bit wide output coming from ALU controller to the PISO. This enables the PISO to send out the output serially.
add_sub_L, add_sub_R	Output	add_sub_L and add_sub_R is 1 bit wide coming from the ALU controller based on the MSB of the coefficient value read from memory. This bit decides whether addition or subtraction that has to be performed between the operands of the adder_L.

Verilog Code:

```
module alu_controller (
       input compute_enable,
       input Clear,
       input Sclk,
       input sleep_flag,
```

```
	input [15:0] rj_data_L, coeff_data_L, xin_data_L,
	input [15:0] rj_data_R, coeff_data_R, xin_data_R,

	output [39:0] add_inp_L, add_inp_R,

	output reg [3:0] rj_addr_L,
	output reg [8:0] coeff_addr_L,
	output reg [7:0] xin_addr_L,

	output reg [3:0] rj_addr_R,
	output reg [8:0] coeff_addr_R,
	output reg [7:0] xin_addr_R,

	output reg rj_en_L, coeff_en_L, xin_en_L,
	output reg rj_en_R, coeff_en_R, xin_en_R,

	output reg add_sub_L, adder_en_L, shift_enable_L, load_L, clear_L, p2s_enable_L,
	output reg add_sub_R, adder_en_R, shift_enable_R, load_R, clear_R, p2s_enable_R
	);

	parameter initial_state = 2'b00, comp_state = 2'b01, sleep_state = 2'b10;

	reg [1:0] pr_state_L, next_state_L;
	reg [1:0] pr_state_R, next_state_R;

	reg [7:0] x_count_L, x_index_L;
	reg [7:0] x_count_R, x_index_R;

	reg [7:0] k_L, k_R;

	reg xmem_overflow_L, start_comp_L, compute_status_L, out_done_L;
	reg xmem_overflow_R, start_comp_R, compute_status_R, out_done_R;

	//wire [39:0] shifted_L, shifted_R, sum_L, sum_R;

	assign add_inp_L = (xin_data_L[15]) ? {8'hFF, xin_data_L, 16'h0000} : {8'h00,
xin_data_L, 16'h0000};
	assign add_inp_R = (xin_data_R[15]) ? {8'hFF, xin_data_R, 16'h0000} : {8'h00,
xin_data_R, 16'h0000};


	always @(Clear, next_state_L)
	begin
		if (Clear == 1'b1)
			pr_state_L <= initial_state;
		else
			pr_state_L <= next_state_L;
	end

	always @(posedge Sclk)
	begin
		//next_state_L <= initial_state;
		case (pr_state_L)
			initial_state:
				begin
					xmem_overflow_L <= 1'b0;
					//out_done_L = 1'b0;
					if (Clear == 1'b1)
						next_state_L <= initial_state;
					else if (compute_enable == 1'b1)
					begin
						next_state_L <= comp_state;
						x_count_L <= 8'd1;
						start_comp_L <= 1'b1;
						compute_status_L <= 1'b1;
```

```
                                end
                                else
                                begin
                                        next_state_L <= initial_state;
                                        x_count_L <= x_count_L;
                                        start_comp_L <= 1'b0;
                                end
                        end

                comp_state:
                        begin
                                if (compute_enable == 1'b1)
                                begin
                                        x_count_L <= x_count_L + 1'b1;
                                        start_comp_L <= 1'b1;
                                        compute_status_L <= 1'b1;
                                        if (x_count_L == 8'hFF)
                                                xmem_overflow_L <= 1'b1;
                                        else
                                                xmem_overflow_L <= xmem_overflow_L;
                                end
                                else
                                begin
                                        start_comp_L <= 1'b0;
                                        xmem_overflow_L <= xmem_overflow_L;
                                        if (rj_addr_L == 4'hF && coeff_addr_L == 9'h1FF
&& k_L == rj_data_L)
                                                compute_status_L <= 1'b0;
                                        else
                                                compute_status_L <= compute_status_L;
                                end

                                if (Clear == 1'b1)
                                        next_state_L <= initial_state;
                                else if (sleep_flag == 1'b1)
                                        next_state_L <= sleep_state;
                                else
                                        next_state_L <= comp_state;
                        end

                sleep_state:
                        begin
                                x_count_L <= x_count_L;
                                xmem_overflow_L <= xmem_overflow_L;
                                start_comp_L <= 1'b0;
                                compute_status_L <= 1'b0;
                                if (Clear == 1'b1)
                                        next_state_L <= initial_state;
                                else if (sleep_flag == 1'b0)
                                begin
                                        x_count_L <= x_count_L + 1'b1;
                                        start_comp_L <= 1'b1;
                                        compute_status_L <= 1'b1;
                                        if (x_count_L == 8'hFF)
                                                xmem_overflow_L <= 1'b1;
                                        else
                                                xmem_overflow_L <= xmem_overflow_L;
                                        next_state_L <= comp_state;
                                end
                                else
                                        next_state_L <= sleep_state;
                        end

                default:        next_state_L <= initial_state;
        endcase
end
```

```
always @(posedge Sclk)
begin
	if (out_done_L)
	begin
		p2s_enable_L = 1'b1;
		rj_addr_L = 4'd0;
		coeff_addr_L = 9'd0;
		k_L = 8'd0;
		out_done_L = 1'b0;
		clear_L = 1'b1;
	end
	else
		p2s_enable_L = 1'b0;

	if (start_comp_L == 1'b1)
	begin
		out_done_L = 1'b0;
		rj_addr_L = 4'd0;
		rj_en_L = 1'b1;
		coeff_addr_L = 9'd0;
		coeff_en_L = 1'b1;
		xin_en_L = 1'b0;
		adder_en_L = 1'b0;
		shift_enable_L = 1'b0;
		k_L = 8'd0;
		clear_L = 1'b1;
		load_L = 1'b0;
	end
	else if (compute_status_L == 1'b1)
	begin
		if (k_L == rj_data_L)
		begin
			xin_en_L = 1'b0;
			shift_enable_L = 1'b1;
			clear_L = 1'b0;
			load_L = 1'b1;
			adder_en_L = 1'b1;
			k_L = 8'd0;
			if (rj_addr_L < 4'd15)
			begin
				rj_addr_L = rj_addr_L + 1'b1;
			end
			else
			begin
				rj_addr_L = 4'd0;
				out_done_L = 1'b1;
				coeff_addr_L = 9'd0;
			end
		end
		else
		begin
			shift_enable_L = 1'b0;
			clear_L = 1'b0;
			load_L = 1'b0;
			xin_en_L = 1'b0;
			x_index_L = coeff_data_L[7:0];
			add_sub_L = coeff_data_L[8];
			if (x_count_L - 1'b1 >= x_index_L)
			begin
				xin_addr_L = x_count_L - 1'b1 - x_index_L;
				xin_en_L = 1'b1;
				adder_en_L = 1'b1;
				load_L = 1'b1;
			end
```

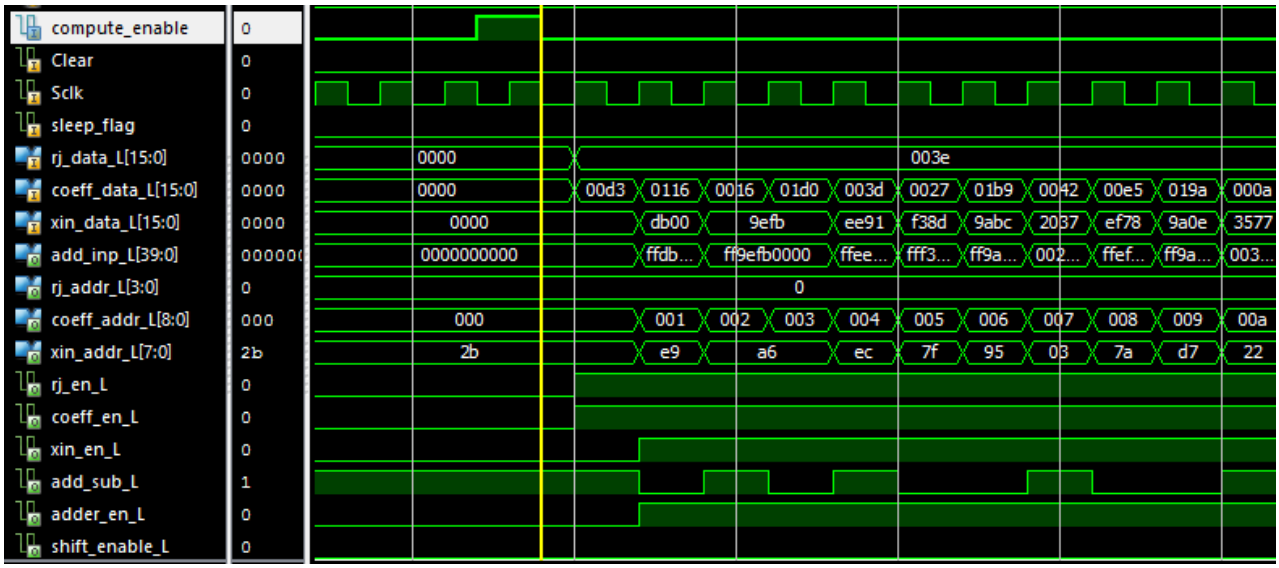
```
                        else if (x_count_L - 1'b1 < x_index_L && xmem_overflow_L ==
1'b1)
                        begin
                                xin_addr_L = x_count_L - 1'b1 + (9'd256 - x_index_L);
                                xin_en_L = 1'b1;
                                adder_en_L = 1'b1;
                                load_L = 1'b1;
                        end
                        else
                        begin
                                xin_addr_L = xin_addr_L;
                                adder_en_L = 1'b0;
                        end

                        if (coeff_addr_L < 9'h1FF)
                                coeff_addr_L = coeff_addr_L + 1'b1;
                        else
                                coeff_addr_L = coeff_addr_L;

                        k_L = k_L + 1'b1;
                end
        end
        else
        begin
                rj_addr_L = 4'd0;
                rj_en_L = 1'b0;
                coeff_addr_L = 9'd0;
                coeff_en_L = 1'b0;
                xin_en_L = 1'b0;
                adder_en_L = 1'b0;
                shift_enable_L = 1'b0;
                k_L = 8'd0;
                load_L = 1'b0;
                clear_L = 1'b1;
        end
end

/*always @ (negedge p2s_enable_L)
begin
                $display("%d : %X \n",x_count_L,shifted_L);
end*/


// Right side FSM

always @(Clear, next_state_R)
begin
        if (Clear == 1'b1)
                pr_state_R <= initial_state;
        else
                pr_state_R <= next_state_R;
end

always @(posedge Sclk)
begin
        //next_state_R <= initial_state;
        case (pr_state_R)
                initial_state:
                        begin
                                xmem_overflow_R <= 1'b0;
                                //out_done_R = 1'b0;
                                if (Clear == 1'b1)
                                        next_state_R <= initial_state;
                                else if (compute_enable == 1'b1)
                                begin
                                        next_state_R <= comp_state;
```

```
                                        x_count_R <= 8'd1;
                                        start_comp_R <= 1'b1;
                                        compute_status_R <= 1'b1;
                                end
                                else
                                begin
                                        next_state_R <= initial_state;
                                        x_count_R <= x_count_R;
                                        start_comp_R <= 1'b0;
                                end
                        end

                comp_state:
                        begin
                                if (compute_enable == 1'b1)
                                begin
                                        x_count_R <= x_count_R + 1'b1;
                                        start_comp_R <= 1'b1;
                                        compute_status_R <= 1'b1;
                                        if (x_count_R == 8'hFF)
                                                xmem_overflow_R <= 1'b1;
                                        else
                                                xmem_overflow_R <= xmem_overflow_R;
                                end
                                else
                                begin
                                        start_comp_R <= 1'b0;
                                        xmem_overflow_R <= xmem_overflow_R;
                                        if (rj_addr_R == 4'hF && coeff_addr_R == 9'h1FF
&& k_R == rj_data_R)
                                                compute_status_R <= 1'b0;
                                        else
                                                compute_status_R <= compute_status_R;
                                end

                                if (Clear == 1'b1)
                                        next_state_R <= initial_state;
                                //else if (sleep_flag == 1'b1)
                                //      next_state_R <= sleep_state;
                                else
                                        next_state_R <= comp_state;
                        end

                sleep_state:
                        begin
                                x_count_R <= x_count_R;
                                xmem_overflow_R <= xmem_overflow_R;
                                start_comp_R <= 1'b0;
                                compute_status_R <= 1'b0;
                                if (Clear == 1'b1)
                                        next_state_R <= initial_state;
                                else if (sleep_flag == 1'b0)
                                begin
                                        x_count_R <= x_count_R + 1'b1;
                                        start_comp_R <= 1'b1;
                                        compute_status_R <= 1'b1;
                                        if (x_count_R == 8'hFF)
                                                xmem_overflow_R <= 1'b1;
                                        else
                                                xmem_overflow_R <= xmem_overflow_R;
                                        next_state_R <= comp_state;
                                end
                                else
                                        next_state_R <= sleep_state;
                        end
```

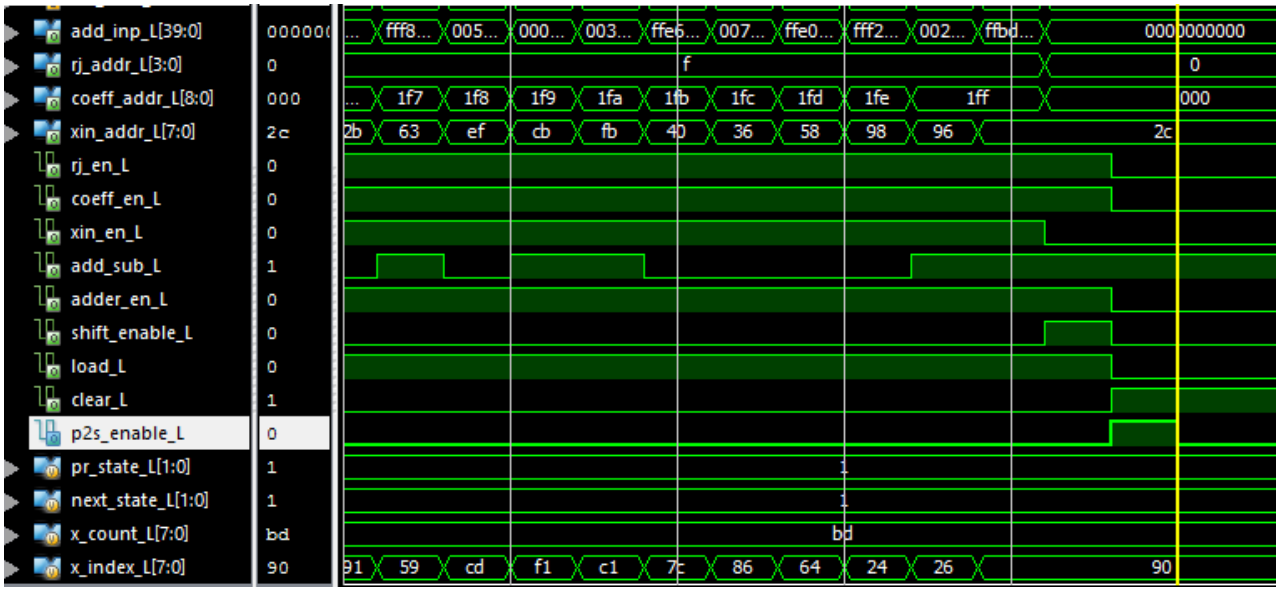
```
		default:
			begin
			end
	endcase
end

always @(posedge Sclk)
begin
	if (out_done_R)
	begin
		p2s_enable_R = 1'b1;
		rj_addr_R = 4'd0;
		coeff_addr_R = 9'd0;
		k_R = 8'd0;
		out_done_R = 1'b0;
	end
	else
		p2s_enable_R = 1'b0;

	if (start_comp_R == 1'b1)
	begin
		out_done_R = 1'b0;
		rj_addr_R = 4'd0;
		rj_en_R = 1'b1;
		coeff_addr_R = 9'd0;
		coeff_en_R = 1'b1;
		xin_en_R = 1'b0;
		adder_en_R = 1'b0;
		shift_enable_R = 1'b0;
		k_R = 8'd0;
		clear_R = 1'b1;
		load_R = 1'b0;
	end
	else if (compute_status_R == 1'b1)
	begin
		if (k_R == rj_data_R)
		begin
			xin_en_R = 1'b0;
			shift_enable_R = 1'b1;
			clear_R = 1'b0;
			load_R = 1'b1;
			adder_en_R = 1'b1;
			k_R = 8'd0;
			if (rj_addr_R < 4'd15)
			begin
				rj_addr_R = rj_addr_R + 1'b1;
			end
			else
			begin
				rj_addr_R = 4'd0;
				out_done_R = 1'b1;
				coeff_addr_R = 9'd0;
			end
		end
		else
		begin
			shift_enable_R = 1'b0;
			clear_R = 1'b0;
			load_R = 1'b0;
			xin_en_R = 1'b0;
			x_index_R = coeff_data_R[7:0];
			add_sub_R = coeff_data_R[8];
			if (x_count_R - 1'b1 >= x_index_R)
			begin
				xin_addr_R = x_count_R - 1'b1 - x_index_R;
				xin_en_R = 1'b1;
```

```
                                adder_en_R = 1'b1;
                                load_R = 1'b1;
                        end
                        else if (x_count_R - 1'b1 < x_index_R && xmem_overflow_R ==
1'b1)
                        begin
                                xin_addr_R = x_count_R - 1'b1 + (9'd256 - x_index_R);
                                xin_en_R = 1'b1;
                                adder_en_R = 1'b1;
                                load_R = 1'b1;
                        end
                        else
                        begin
                                xin_addr_R = xin_addr_R;
                                adder_en_R = 1'b0;
                        end

                        if (coeff_addr_R < 9'h1FF)
                                coeff_addr_R = coeff_addr_R + 1'b1;
                        else
                                coeff_addr_R = coeff_addr_R;

                        k_R = k_R + 1'b1;
                end
        end
        else
        begin
                rj_addr_R = 4'd0;
                rj_en_R = 1'b0;
                coeff_addr_R = 9'd0;
                coeff_en_R = 1'b0;
                xin_en_R = 1'b0;
                adder_en_R = 1'b0;
                shift_enable_R = 1'b0;
                k_R = 8'd0;
                load_R = 1'b0;
                clear_R = 1'b1;
        end
    end

    /*always @ (negedge p2s_enable_R)
    begin
                $display("%d : %X \n",x_count_R,shifted_R);
    end*/

endmodule
```

Simulation Results:



Here we see the computation beginning with the compute_enable signal asserted and the entire rj, coeff and data streams beginning to be fetched – memories are being activated, values are checked and add/subtracts and shifts carried out.



In the above diagram, we see the signal p2s_enable_L being asserted which signals to the PISO block that the data at the shift_acc block output is valid as output for this computation iteration.

8. MSDAP Controller:

Functionality:

Signal	Input/output	Description
Sclk	Input	Sclk is the system clock that is used to synchronize all the control signals and IO signals. All the state changes and the computation happen with the negative or falling edge of the System clock. The frequency of the Sclk determines the frequency of operation of the entire system/ chip. This is an input signal to the MSDAP from the controller
Dclk	Input	Dclk is the Data clock that is used as the timing reference to receive and send the input samples. A bit is received or sent at the falling edge of the Dclk. Hence to receive a 16 bit data, 16 Dclk cycles are required. One Dclk is 35 Sclk cycles. This is an input signal to the MSDAP from the controller
Frame	Input	Frame is an input signal from the controller to the MSDAP. While sending or receiving the input or output to or from the MSDAP, the Frame is set high. Frame signifies the start of the first bit of data-input, output, Rj or co-efficient. Frame is used as active high signal and is made high for one Dclk cycle. After receiving the first bit of the data, the Frame goes inactive i.e. low. Frame is made active after every 16 Dclk cycles and hence a 16 bit data is received between two successive Frames.
Start	Input	Start is an asynchronous input signal from the Controller to MSDAP. It is used to indicate the start of the FSM and initialization of chip begins. Until then the chip is said to be idle. Start is an active high input signal.

Reset_n	Input	Reset is an active low asynchronous input signal to the MSDAP. The chip goes to the reset mode when this signal is applied.
input_rdy_flag	Input	input_rdy_flag signals to the controller that the input data is ready for processing
zero_flag_L, zero_flag_R	Input	Signals from the data memory block that indicate that there have been 800 zeroes detected and both flags are ANDed to generate sleep_flag
rj_wr_addr, coeff_wr_addr, data_wr_addr	Output	These signals are used for signaling the memories of the addresses where the incoming data must be written.
rj_en, coeff_en, data_en	Output	Enables for routing the data to the correct memories as it comes in.
Frame_out, Dclk_out, Sclk_out	Output	Internal signals for Frame, Dclk and Sclk, distributed from the main controller
compute_enable	Output	Indicates to the ALU that a computation iteration can begin
sleep_flag	Output	Signals to the ALU that the main controller will be in sleep mode and that computations must cease
InReady	Output	InReady is an output signal to the Controller from the MSDAP which is set high when the chip is ready to receive the data from both the channels. InReady is aligned with the Frame.

Verilog Code:

```
module MSDAP_controller (input Sclk, Dclk, Start, Reset_n, Frame, input_rdy_flag,
zero_flag_L, zero_flag_R,
                                                    output reg [3:0] rj_wr_addr,
                                                output reg [8:0] coeff_wr_addr,
                                                output reg [7:0] data_wr_addr,
                                                output  reg  rj_en,  coeff_en,  data_en,
Clear,
                                                output Frame_out, Dclk_out, Sclk_out,
                                                output  reg  compute_enable,  sleep_flag,
InReady);
```

```
	parameter [3:0] Startup = 4'd0, Wait_rj = 4'd1, Read_rj = 4'd2,
	                                Wait_coeff = 4'd3, Read_coeff = 4'd4, Wait_input =
4'd5,
	                                Compute = 4'd6, Reset = 4'd7, Sleep = 4'd8;

	reg [3:0] pr_state, next_state;
	reg [15:0] real_count;

	reg [4:0] rj_count;
	reg [9:0] coeff_count;
	reg [7:0] data_count;

	reg taken;

	assign Frame_out = Frame;
	assign Dclk_out = Dclk;
	assign Sclk_out = Sclk;

	always @(negedge Sclk or negedge Reset_n)            // Sequential block
	begin
		if (!Reset_n)
		begin
			if (pr_state > 4'd4)
				pr_state = Reset;
			else
				pr_state = next_state;
		end
		else
		pr_state = next_state;
	end

	always @(negedge Sclk or posedge Start)
	begin
		if (Start == 1'b1)
			next_state = Startup;
		else
		begin
		case (pr_state)
			Startup:	begin
						rj_wr_addr = 4'd0;
						coeff_wr_addr = 9'd0;
						data_wr_addr = 8'd0;
						rj_en = 1'b0;
						coeff_en = 1'b0;
						data_en = 1'b0;
						Clear = 1'b1;
						compute_enable = 1'b0;
						InReady = 1'b0;
						sleep_flag = 1'b0;
						next_state = Wait_rj;
						real_count = 16'd0;
						rj_count = 4'd0;
						coeff_count = 9'd0;
						data_count = 8'd0;
					end

			Wait_rj:	begin
						rj_wr_addr = 4'd0;
						coeff_wr_addr = 9'd0;
						data_wr_addr = 8'd0;
						rj_en = 1'b0;
						coeff_en = 1'b0;
						data_en = 1'b0;
						Clear = 1'b0;
						compute_enable = 1'b0;
						InReady = 1'b1;
```

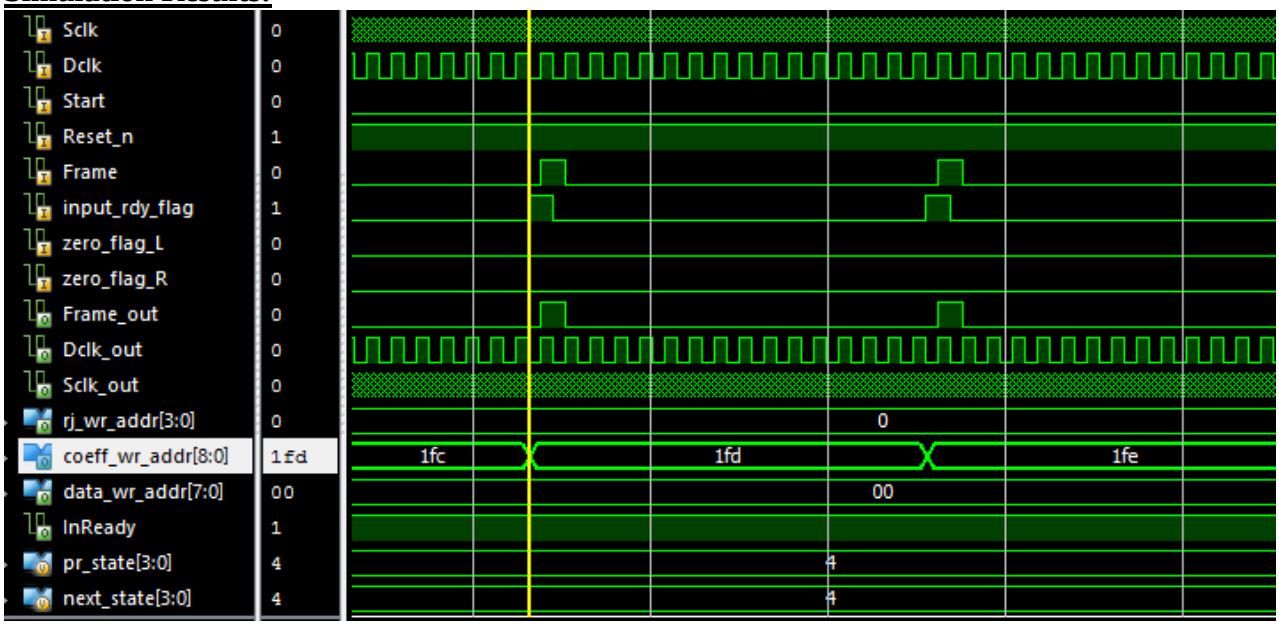
```
                                        sleep_flag = 1'b0;
                                        rj_count = 4'd0;
                                        coeff_count = 9'd0;
                                        data_count = 8'd0;
                                        taken = 1'b0;
                                        if (Frame == 1'b1)
                                                next_state = Read_rj;
                                        else
                                                next_state = Wait_rj;
                                end

                        Read_rj:        begin
                                        coeff_wr_addr = 9'd0;
                                        data_wr_addr = 8'd0;
                                        coeff_en = 1'b0;
                                        data_en = 1'b0;
                                        Clear = 1'b0;
                                        compute_enable = 1'b0;
                                        InReady = 1'b1;
                                        sleep_flag = 1'b0;
                                        coeff_count = 9'd0;
                                        data_count = 8'd0;
                                        if (input_rdy_flag == 1'b1 && taken ==
1'b0)
                                        begin
                                                if (rj_count < 5'd16)
                                                begin
                                                        rj_en = 1'b1;
                                                        rj_wr_addr = rj_count;
                                                        rj_count = rj_count + 1'b1;
                                                        next_state = Read_rj;
                                                        taken = 1'b1;
                                                end
                                                if (rj_count == 5'd16)
                                                begin
                                                        next_state = Wait_coeff;
                                                end
                                                else
                                                        next_state = Read_rj;
                                        end
                                        else if (input_rdy_flag == 1'b0)
                                        begin
                                                taken = 1'b0;
                                                rj_en = 1'b0;
                                                rj_wr_addr = rj_wr_addr;
                                                next_state = Read_rj;
                                        end
                                        else
                                                next_state = Read_rj;
                                end

                        Wait_coeff:
                                        begin
                                                rj_wr_addr = 4'd0;
                                                coeff_wr_addr = 9'd0;
                                                data_wr_addr = 8'd0;
                                                rj_en = 1'b0;
                                                coeff_en = 1'b0;
                                                data_en = 1'b0;
                                                Clear = 1'b0;
                                                compute_enable = 1'b0;
                                                InReady = 1'b1;
                                                sleep_flag = 1'b0;
                                                coeff_count = 9'd0;
                                                data_count = 8'd0;
                                                if (Frame == 1'b1)
```

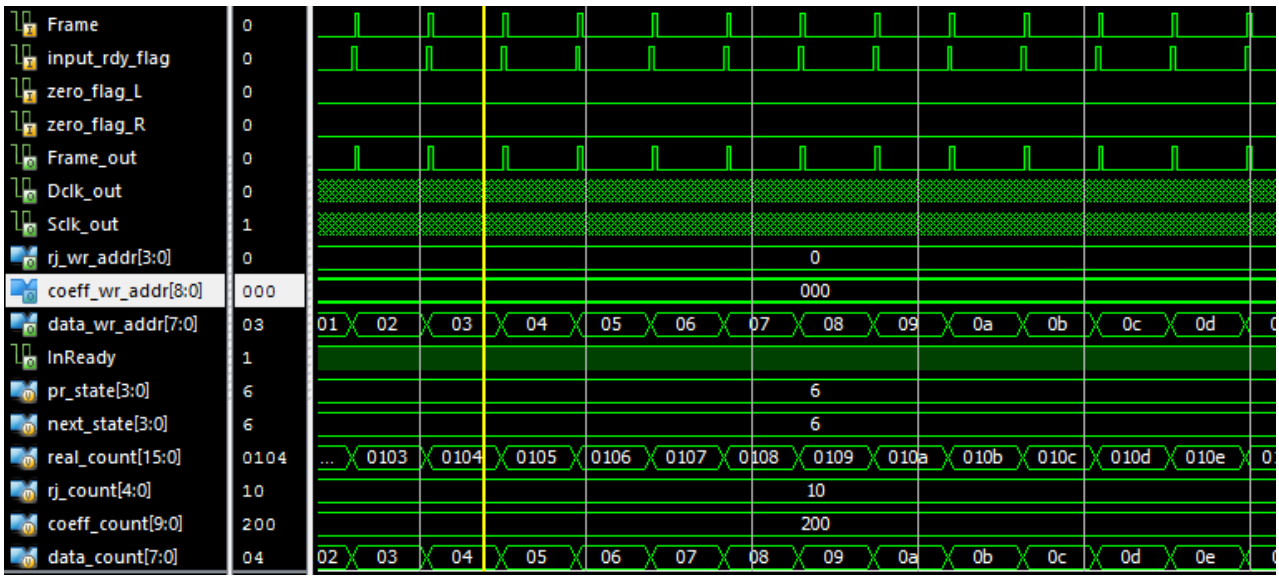
```
                        next_state = Read_coeff;
                else
                        next_state = Wait_coeff;
        end

        Read_coeff: begin
                rj_wr_addr = 4'd0;
                data_wr_addr = 8'd0;
                rj_en = 1'b0;
                data_en = 1'b0;
                Clear = 1'b0;
                compute_enable = 1'b0;
                InReady = 1'b1;
                sleep_flag = 1'b0;
                data_count = 8'd0;
                if (input_rdy_flag == 1'b1 && taken
== 1'b0)
                begin
                        if (coeff_count < 10'h200)
                        begin
                                coeff_en = 1'b1;
                                coeff_wr_addr        =
coeff_count;
                                coeff_count          =
coeff_count + 1'b1;
                                next_state           =
Read_coeff;
                                taken = 1'b1;
                        end
                        if (coeff_count == 10'h200)
                                next_state           =
Wait_input;
                        else
                                next_state           =
Read_coeff;
                end
                else if (input_rdy_flag == 1'b0)
                begin
                        taken = 1'b0;
                        coeff_en = 1'b0;
                        coeff_wr_addr                =
coeff_wr_addr;
                        next_state = Read_coeff;
                end
                else
                        next_state = Read_coeff;
        end

        Wait_input: begin
                rj_wr_addr = 4'd0;
                coeff_wr_addr = 9'd0;
                data_wr_addr = 8'd0;
                rj_en = 1'b0;
                coeff_en = 1'b0;
                data_en = 1'b0;
                Clear = 1'b0;
                compute_enable = 1'b0;
                InReady = 1'b1;
                sleep_flag = 1'b0;
                data_count = 8'd0;
                if (Reset_n == 1'b0)
                        next_state = Reset;
                else if (Frame == 1'b1)
                        next_state = Compute;
                else
                        next_state = Wait_input;
```

```
                                        end

                        Compute:        begin
                                        rj_wr_addr = 4'd0;
                                        coeff_wr_addr = 9'd0;
                                        rj_en = 1'b0;
                                        coeff_en = 1'b0;
                                        Clear = 1'b0;
                                        InReady = 1'b1;
                                        sleep_flag = 1'b0;
                                        if (Reset_n == 1'b0)
                                        begin
                                               Clear = 1'b1;
                                               next_state = Reset;

                                        end
                                        else if (input_rdy_flag == 1'b1 && taken
== 1'b0)
                                        begin
                                               if (zero_flag_L && zero_flag_R)
                                               begin
                                                      next_state = Sleep;
                                                      sleep_flag = 1'b1;
                                               end
                                               else
                                               begin
                                                      data_en = 1'b1;
                                                      data_wr_addr = data_count;
                                                      data_count  =  data_count  +
1'b1;
                                                      real_count  =  real_count  +
1'b1;
                                                      next_state = Compute;
                                                      compute_enable = 1'b1;
                                                      taken = 1'b1;
                                               end
                                        end
                                        else if (input_rdy_flag == 1'b0)
                                        begin
                                               taken = 1'b0;
                                               data_en = 1'b0;
                                               data_wr_addr = data_wr_addr;
                                               compute_enable = 1'b0;
                                               next_state = Compute;
                                        end
                                        else
                                        begin
                                               data_en = 1'b0;
                                               data_wr_addr = data_wr_addr;
                                               //real_count = real_count + 1'b1;
                                               next_state = Compute;
                                               compute_enable = 1'b0;
                                        end
                                end

                        Reset: begin
                                        rj_wr_addr = 4'd0;
                                        coeff_wr_addr = 9'd0;
                                        data_wr_addr = 8'd0;
                                        rj_en = 1'b0;
                                        coeff_en = 1'b0;
                                        data_en = 1'b0;
                                        Clear = 1'b1;
                                        compute_enable = 1'b0;
                                        InReady = 1'b0;
                                        sleep_flag = 1'b0;
```

```
                                        data_count = 8'd0;
                                        taken = 1'b0;
                                        //real_count = real_count - 1'b1;
                                        if (Reset_n == 1'b0)
                                               next_state = Reset;
                                        else
                                               next_state = Wait_input;
                                 end

                        Sleep: begin
                                        rj_wr_addr = 4'd0;
                                        coeff_wr_addr = 9'd0;
                                        data_wr_addr = data_wr_addr;
                                        rj_en = 1'b0;
                                        coeff_en = 1'b0;
                                        data_en = 1'b0;
                                        Clear = 1'b0;
                                        compute_enable = 1'b0;
                                        InReady = 1'b1;
                                        sleep_flag = 1'b1;
                                        if (Reset_n == 1'b0)
                                               next_state = Reset;
                                        else if (input_rdy_flag == 1'b1 && taken
== 1'b0)
                                        begin
                                               if (zero_flag_L && zero_flag_R)
                                                      next_state = Sleep;
                                               else
                                               begin
                                                      taken = 1'b1;
                                                      data_en = 1'b1;
                                                      compute_enable = 1'b1;
                                                      sleep_flag = 1'b0;
                                                      data_wr_addr = data_count;
                                                      data_count  =  data_count  +
1'b1;
                                                      real_count  =  real_count  +
1'b1;
                                                      next_state = Compute;
                                               end
                                        end
                                        else
                                               next_state = Sleep;
                                 end

                endcase
                end
        end
endmodule
```

Simulation Results:





In the above two diagrams, we see the operation of the Main controller, with inputs being continuously read and written to memory. Also, compute_enable signals are issued to start the computations in the ALU.

9. Parallel to Serial Output [P2S]:

Functionality:

This functional block operates on the rising edge of the Sclk and the FRAME. Parallel data starting from the MSB of the 16bits input data is received at the arrival of frame and it is captured in the rising edge of the Sclk. Once the frame goes high, the MSB data is received on that edge and the rest 15 bits of data is received at every Sclk positive edge.

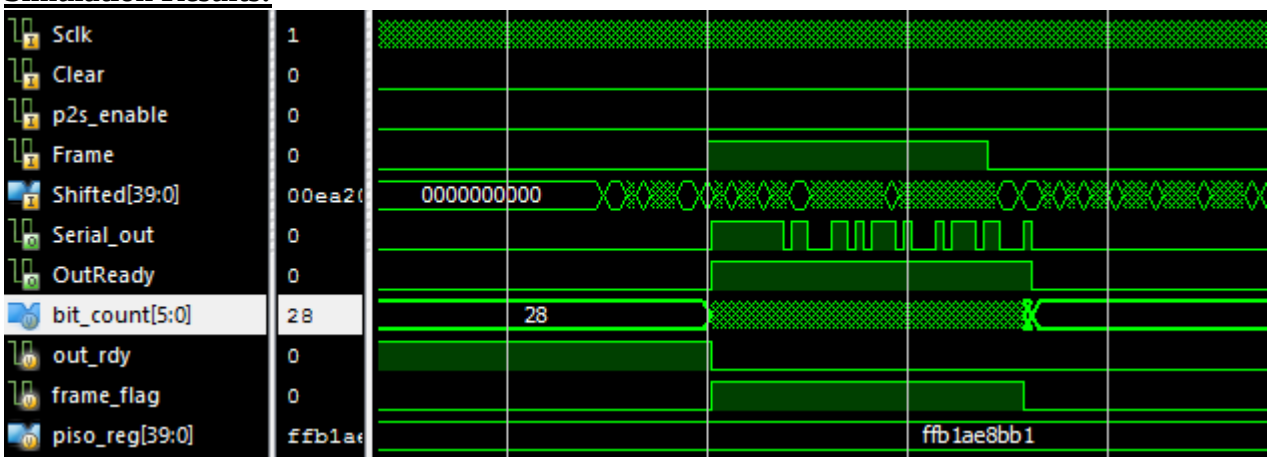
Signal	Input/output	Description
Clear	Input	Clear is synchronous with Sclk. Upon receiving clear signal from the MSDAP Controller, data in the P2S is cleared.
Sclk	Input	Sclk is the timing reference for receiving the input data parallel. Data is received parallel at the positive edge of the Sclk.
Shifted_L, Shifted_R	Input	Shifted_L, Shifted_R are the parallel inputs to the P2S (left and right respectively) from the shift_acc_L and shift_acc_R. These input data are captured on the positive edge of Sclk.
Output_L, Output_R	Output	Output_L, output_R is the serial output from P2S left and right blocks respectively.
p2s_enable_L, p2s_enable_R	Input	This is one bit from the ALU controller to the P2S left and right block respectively. This is set when the output is ready and has to be sent out serially.

Verilog Code:

```
module PISO (Sclk, Clear, Frame, Shifted, Serial_out, p2s_enable, OutReady);
input Sclk, Clear, p2s_enable, Frame;
input [39:0] Shifted;
output reg Serial_out, OutReady;
reg [5:0] bit_count;

reg out_rdy, frame_flag;
reg [39:0] piso_reg;

	always @(negedge Sclk)
	begin
		if(Clear == 1'b1)
		begin
			bit_count = 6'd40;
			piso_reg = 40'd0;
			out_rdy = 1'b0;
```

```
                frame_flag = 1'b0;
                OutReady = 1'b0;
                Serial_out = 1'b0;
        end
        else if (p2s_enable == 1'b1)
        begin
                piso_reg = Shifted;
                out_rdy = 1'b1;
        end
        else if (Frame == 1'b1 && out_rdy == 1'b1 && frame_flag == 1'b0)
        begin
                bit_count = bit_count - 1'b1;
                Serial_out = piso_reg [bit_count];
                frame_flag = 1'b1;
                out_rdy = 1'b0;
                OutReady = 1'b1;
        end
        else if (frame_flag == 1'b1)
        begin
                bit_count = bit_count - 1'b1;
                Serial_out = piso_reg [bit_count];
                OutReady = 1'b1;
                if (bit_count == 6'd0)
                        frame_flag = 1'b0;
        end
        else
        begin
                bit_count = 6'd40;
                //piso_reg = 40'd0;
                //out_rdy = 1'b0;
                //frame_flag = 1'b0;
                Serial_out = 1'b0;
                OutReady = 1'b0;
        end
    end
endmodule
```

Simulation Results:

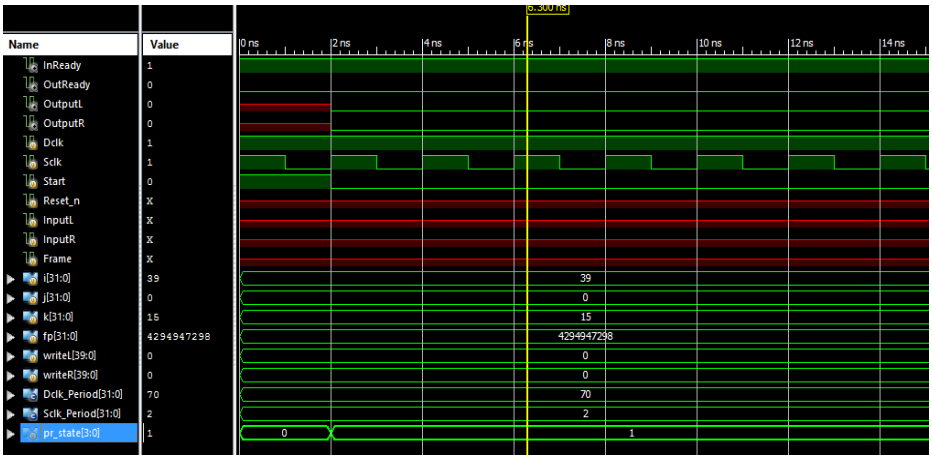


As we can see above, the data FFB1AE8BB1 is stored in the piso_reg and then is released bit by bit as soon as Frame is asserted from Serial_out and OutReady is also asserted when the serial output is going on.

Simulation Results of State transitions

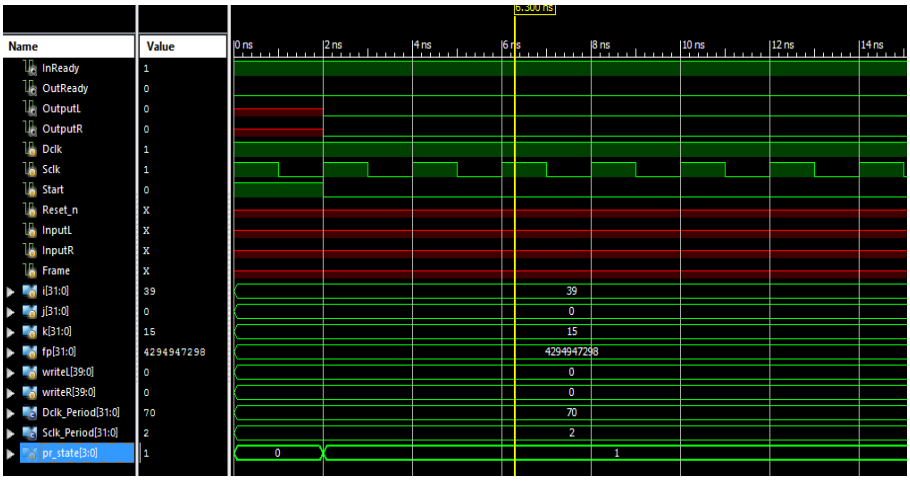
1. Start

The FSM starts to function when it receives a START signal from the controller. Until the MSDAP receives this signal, FSM will be non-functional. Once the MSDAP receives the START signal, the FSM enters into state0 or FSM starts functioning. The below waveform shows the required signals to make the transition to state 0.



2. Transition from State 0 to State 1

- In state0, the '**InReady**' signal is made high.
- All the required temporary registers and flags are initialized in this state. Once this is done, the FSM enters into state 1. This can be seen in the below waveform.



3. Transition from State 1 to State 2

- InReady signal is made high
- In the state 1, the MSDAP is waiting to receive the Rj values.
- When the Rj values for both the channels is about to receive by MSDAP, the controller send a FRAME low, signaling the start of Rj values.
- Once the FRAME is low, the FSM enters in to state 2 to receive the Rj values. Below waveform shows the state transition from 1 to 2 when the FRAME becomes low.

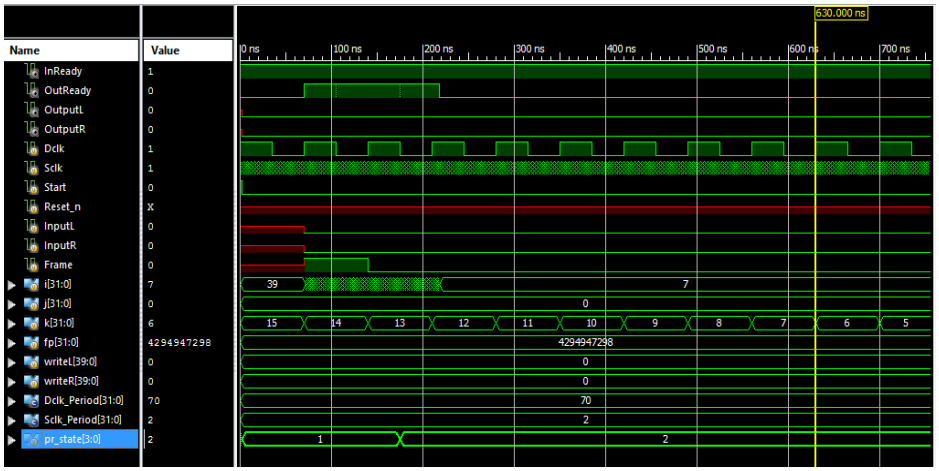
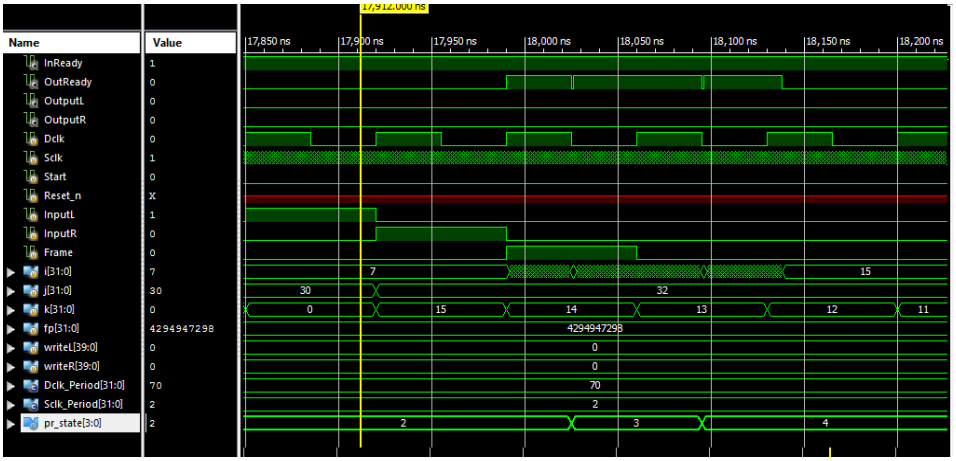


Fig 3

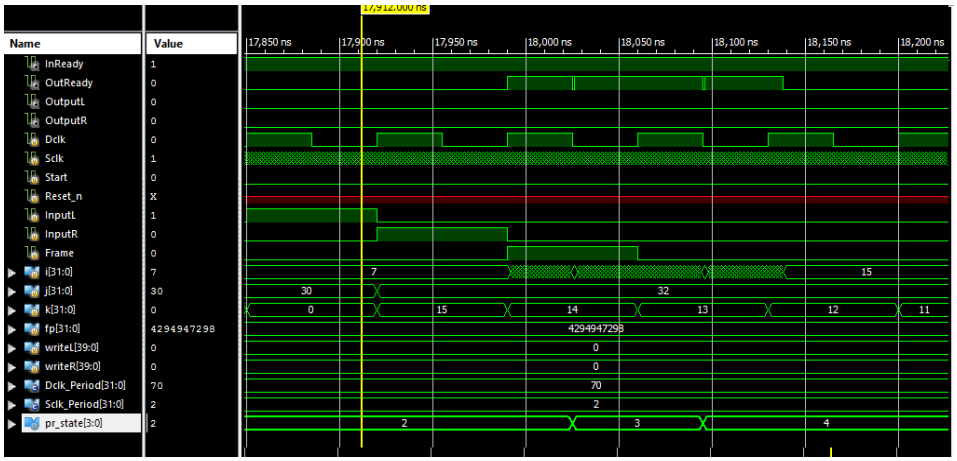
4. Transition from State 2 to State 3

- InReady signal is made high.
- Once all the Rj values are received by the FSM serially and stored in the Rj array, the FSM enters into State 3. All incoming data is through serial bus InputL and InputR from the controller.
- The Rj value is received at the start of State 2 (Fig 4) and after all Rj is received, state is changed to State3.



5. Transition from State 3 to State 4

- InReady signal is made high
- In the state 3, the MSDAP is waiting to receive the Co-efficient values.
- When the Co-efficient values for both the channels is about to receive by MSDAP, the controller send a FRAME low, signaling the start of Co-efficient values.
- Once the FRAME is low, the FSM enters in to state 4 to receive the Coefficient values. Below waveform (Fig 5) shows the state transition from 3 to 4 when the FRAME becomes low.

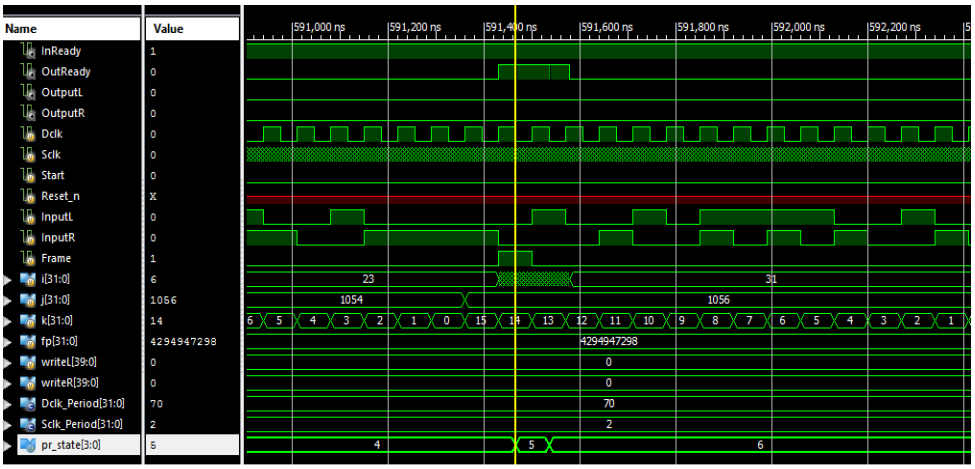


6. Transition from State 4 to State 5

- InReady signal is made high.
- In state 4, the 16 Co-efficient values, each word of length 16 bits is received serially from the controller.
- Once all the Co-efficient values are received by the FSM serially and stored in the Co-efficient array, the FSM enters into State 5.
- The Co-efficient value is received at the start of State 4 and after all Co-efficient is received, state is changed to State 5.

7. Transition from State 5 to State 6

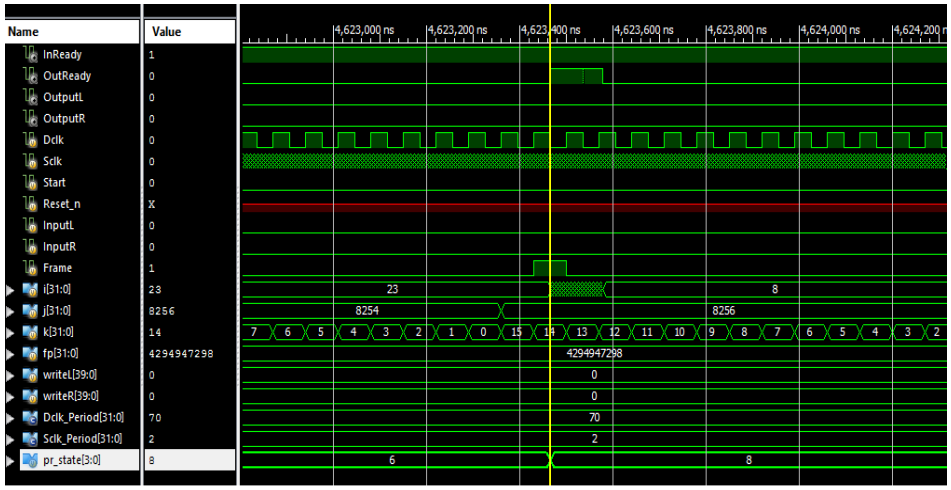
InReady signal is made high.
In state 5, the MSDAP is waiting to receive the data values.
When the data value is about to be received by MSDAP, the controller sends a FRAME low, signaling the start of data values in both the channels.
Once the FRAME is low, the FSM enters in to state 6 to receive the data values. Below waveform shows the state transition from 5 to 6 when the FRAME becomes high.



8. Working in State 6

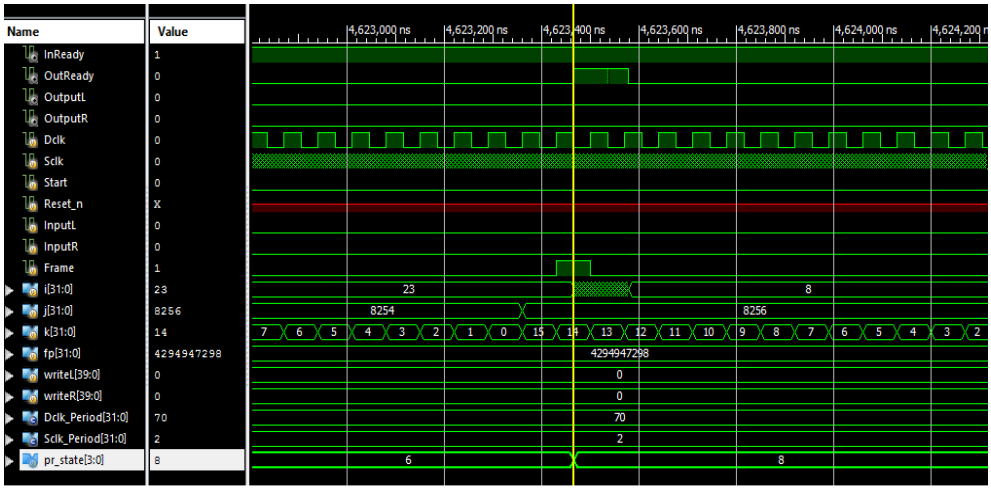
In State 6, the data is being read and the sign is extended to 16 bits and 16 zeroes are concatenated to the LSB.

- When the first frame occurs in state 6, two inputs from InputL and InputR are begin read serially. During the second frame, the computation on the first data is carried and also the second set of data is being received.
- When there is a third frame, the outputs for the first data sample from both the channels are being sent out serially and computation for the second set of data is being carried out. Also, the third set of input data samples is being received by the MSDAP.
- The OutReady goes low whenever the computation is done on the input data samples and the output is sent serially. Once, the 40 bit output sample is sent serially, the OutReady goes high again.
  This operation is shown by the waveforms below:

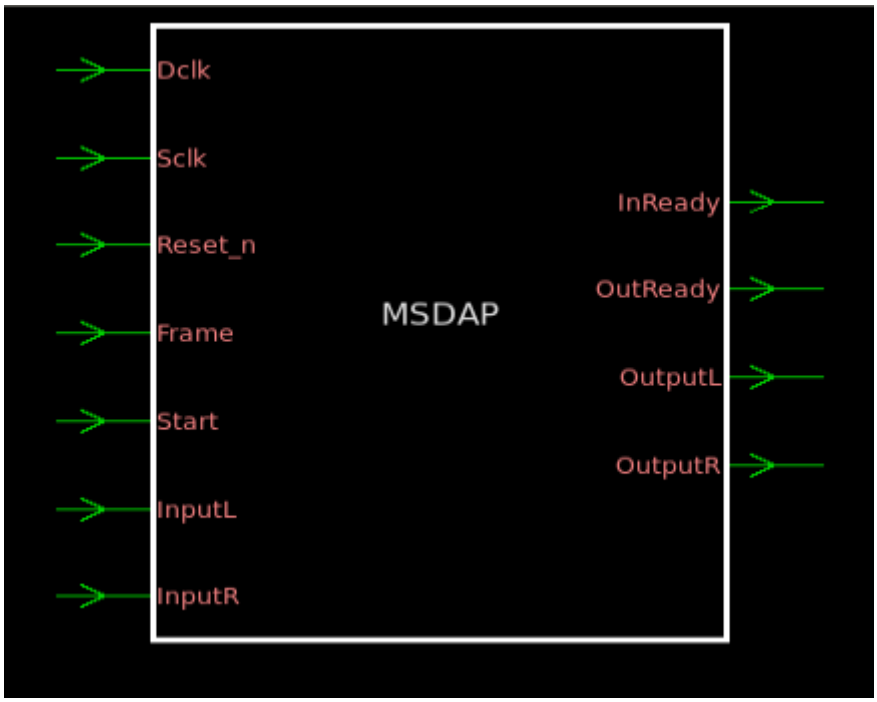


9. Transition from State 6 to State 8

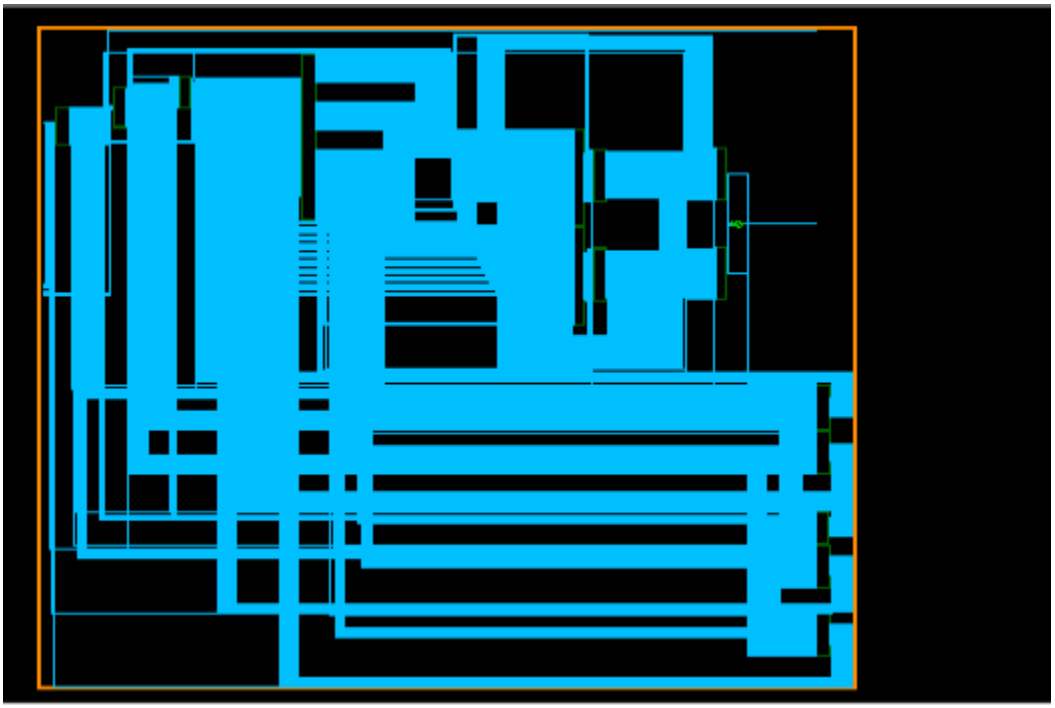
In state 6, if the MSDAP receives 800 consecutive zeroes on both left and right input channels, then the MSDAP will go to sleep mode (State 8). In sleep mode, the OutReady signal goes high. Figure below shows this operation.



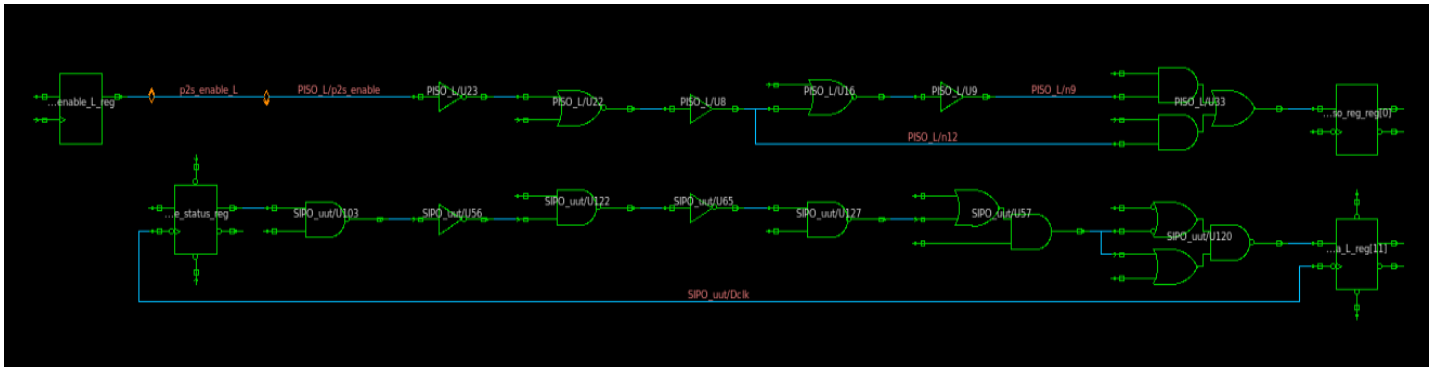
Pin Diagram of Complete MSDAP



Schematic of Complete MSDAP



Critical Path of Complete MSDAP



Design Vision Reports of Gate Level Netlist

1. **Cell Report**

****

Report : cell
Design : MSDAP

Version: H-2013.03-SP4
Date : Wed Dec 16 00:54:06 2015
****

Attributes:
b - black box (unknown)
h - hierarchical
n - noncombinational
r - removable
u - contains unmapped logic

Cell	Reference	Library	Area	Attributes
PISO_L	PISO_0		4418.937628	
PISO_R	PISO_1		4418.937628	
SIPO_uut	SIPO		3938.188885	
U2	INVXLM	ff_1v98_0c	6.585600	
U3	INVX2M	ff_1v98_0c	6.585600	
U4	OR2X2M	ff_1v98_0c	10.976000	
add_L	adder_0		5310.188868	
add_R	adder_1		5310.188868	
alu_ctrl	alu_controller		13015.340654	
coeff_L	coeff_memory_0		847184.757870	
coeff_R	coeff_memory_1		847193.538670	
main_ctrl	MSDAP_controller		5940.211239	
rj_L	rj_memory_0		26252.396887	
rj_R	rj_memory_1		26234.835287	
shift_acc_L	shift_acc_0		2840.588757	
shift_acc_R	shift_acc_1		2840.588757	
xin_L	data_memory_0		426215.642895	
xin_R	data_memory_1		426215.642895	
Total 18 cells			2647354.132984	

2. Area Report

****
Report : area
Design : MSDAP
Version: H-2013.03-SP4
Date : Wed Dec 16 00:54:05 2015
****

Library(s) Used:

ff_1v98_0c (File: /home/eng/x/xxw122030/synopsys/ff_1v98_0c.db)

Number of ports: 11
Number of nets: 476
Number of cells: 18
Number of combinational cells: 3
Number of sequential cells: 0
Number of macros/black boxes: 0
Number of buf/inv: 2
Number of references: 18

Combinational area: 1360879.096340
Buf/Inv area: 105187.396791
Noncombinational area: 1286475.036644
Macro/Black Box area: 0.000000
Net Interconnect area: undefined (No wire load specified)

Total cell area: 2647354.132984
Total area: undefined

3. QOR Report

****
Report : qor
Design : MSDAP
Version: H-2013.03-SP4
Date : Wed Dec 16 00:54:06 2015
****

Timing Path Group 'Dclk'

Levels of Logic: 7.00
Critical Path Length: 0.84
Critical Path Slack: 1301.12
Critical Path Clk Period: 1302.00
Total Negative Slack: 0.00
No. of Violating Paths: 0.00
Worst Hold Violation: 0.00
Total Hold Violation: 0.00
No. of Hold Violations: 0.00

Timing Path Group 'Sclk'

Levels of Logic: 8.00
Critical Path Length: 1.02
Critical Path Slack: 0.02
Critical Path Clk Period: 37.00
Total Negative Slack: 0.00
No. of Violating Paths: 0.00
Worst Hold Violation: 0.00
Total Hold Violation: 0.00
No. of Hold Violations: 0.00

Cell Count

Hierarchical Cell Count: 33
Hierarchical Port Count: 1895
Leaf Cell Count: 83113
Buf/Inv Cell Count: 12441
Buf Cell Count: 10594
Inv Cell Count: 1847
CT Buf/Inv Cell Count: 0
Combinational Cell Count: 57617
Sequential Cell Count: 25496
Macro Count: 0

Area

Combinational Area: 1360879.096340
Noncombinational Area:
1286475.036644
Buf/Inv Area: 105187.396791
Total Buffer Area: 93023.79
Total Inverter Area: 12163.60
Macro/Black Box Area: 0.000000
Net Area: 0.000000

Cell Area: 2647354.132984

Design Area: 2647354.132984

Design Rules

Total Number of Nets: 108581
Nets With Violations: 0
Max Trans Violations: 0
Max Cap Violations: 0

Hostname: engnx12.utdallas.edu

Compile CPU Statistics

Resource Sharing: 42.99
Logic Optimization: 143.14
Mapping Optimization: 242.70

Overall Compile Time: 487.75
Overall Compile Wall Clock Time: 491.77

4. Timing report

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : MSDAP
Version: H-2013.03-SP4
Date   : Wed Dec 16 00:54:05 2015
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: ff_1v98_0c   Library: ff_1v98_0c
Wire Load Model Mode: top

  Startpoint: SIPO_uut/frame_status_reg
              (falling edge-triggered flip-flop clocked by Dclk)
  Endpoint: SIPO_uut/data_L_reg[11]
            (falling edge-triggered flip-flop clocked by Dclk)
```

Path Group: Dclk
Path Type: max

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (ideal)	0.00	651.00
SIPO_uut/frame_status_reg/CKN (DFFNSRHX1M)	0.00	651.00 f
SIPO_uut/frame_status_reg/Q (DFFNSRHX1M)	0.28	651.28 r
SIPO_uut/U103/Y (NAND2X2M)	0.06	651.34 f
SIPO_uut/U56/Y (INVX2M)	0.07	651.41 r
SIPO_uut/U122/Y (NAND2X2M)	0.04	651.45 f
SIPO_uut/U65/Y (INVX2M)	0.05	651.50 r
SIPO_uut/U127/Y (NAND2X2M)	0.04	651.54 f
SIPO_uut/U57/Y (OA21X1M)	0.19	651.73 f
SIPO_uut/U120/Y (OAI2BB2X1M)	0.10	651.84 f
SIPO_uut/data_L_reg[11]/D (DFFNSRHX1M)	0.00	651.84 f
data arrival time	651.84	
clock Dclk (fall edge)	1953.00	1953.00
clock network delay (ideal)	0.00	1953.00
SIPO_uut/data_L_reg[11]/CKN (DFFNSRHX1M)	0.00	1953.00 f
library setup time	-0.04	1952.96
data required time		1952.96
data required time		1952.96
data arrival time		-651.84
slack (MET)		1301.12

Startpoint: SIPO_uut/input_rdy_flag_reg
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: main_ctrl/data_count_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: max

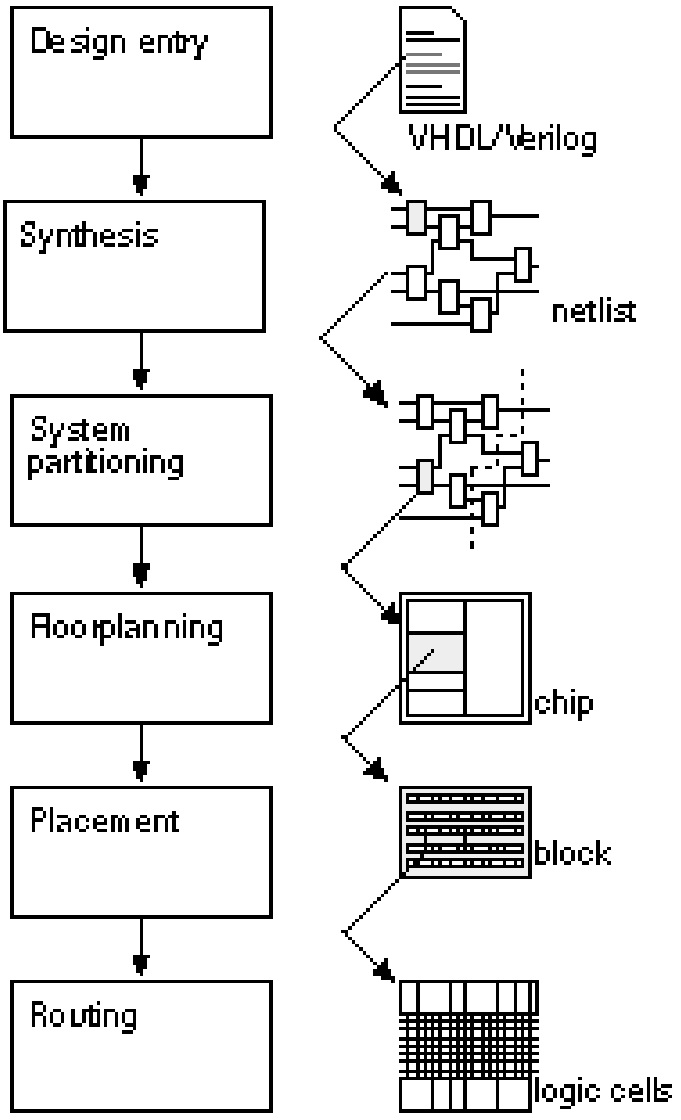
Point	Incr	Path
clock Dclk (fall edge)	26691.00	26691.00
clock network delay (ideal)	0.00	26691.00
SIPO_uut/input_rdy_flag_reg/CKN (DFFNSRHX1M)	0.00	26691.00 f
SIPO_uut/input_rdy_flag_reg/Q (DFFNSRHX1M)	0.30	26691.30 r
SIPO_uut/input_rdy_flag (SIPO)	0.00	26691.30 r

```
main_ctrl/input_rdy_flag (MSDAP_controller)          0.00   26691.30 r
main_ctrl/U28/Y (NAND2XLM)                           0.10   26691.40 f
main_ctrl/U5/Y (NAND2BX2M)                           0.12   26691.52 f
main_ctrl/U27/Y (INVXLM)                             0.08   26691.60 r
main_ctrl/U49/Y (NAND2X2M)                           0.04   26691.64 f
main_ctrl/U12/Y (AOI21X1M)                           0.06   26691.70 r
main_ctrl/U10/Y (BUFX2M)                             0.14   26691.84 r
main_ctrl/U13/Y (AND2X2M)                            0.10   26691.94 r
main_ctrl/U186/Y (OAI2BB2X1M)                        0.08   26692.02 r
main_ctrl/data_count_reg[0]/D (DFFNHX2M)             0.00   26692.02 r
data arrival time                                   26692.02

clock Sclk (fall edge)                          26692.00   26692.00
clock network delay (ideal)                         0.00   26692.00
main_ctrl/data_count_reg[0]/CKN (DFFNHX2M)          0.00   26692.00 f
library setup time                                  0.04   26692.04
data required time                                 26692.04
-----------------------------------------------------------------------
data required time                                 26692.04
data arrival time                                 -26692.02
-----------------------------------------------------------------------
slack (MET)                                         0.02
```

Physical Design Flow

Introduction

Physical Design flow starts right after the circuit design and logical verification of the gate level net list. Gates are converted to represent the actual layers of the component which is later used for fabrication. Gates from the net list are mapped with the equivalent layouts at the end of physical design which gives complete actual reliable information about the critical path, RC delays, power consumption, area and timing. Physical design flow is as shown in the figure below.



Steps involved in Physical Design Flow

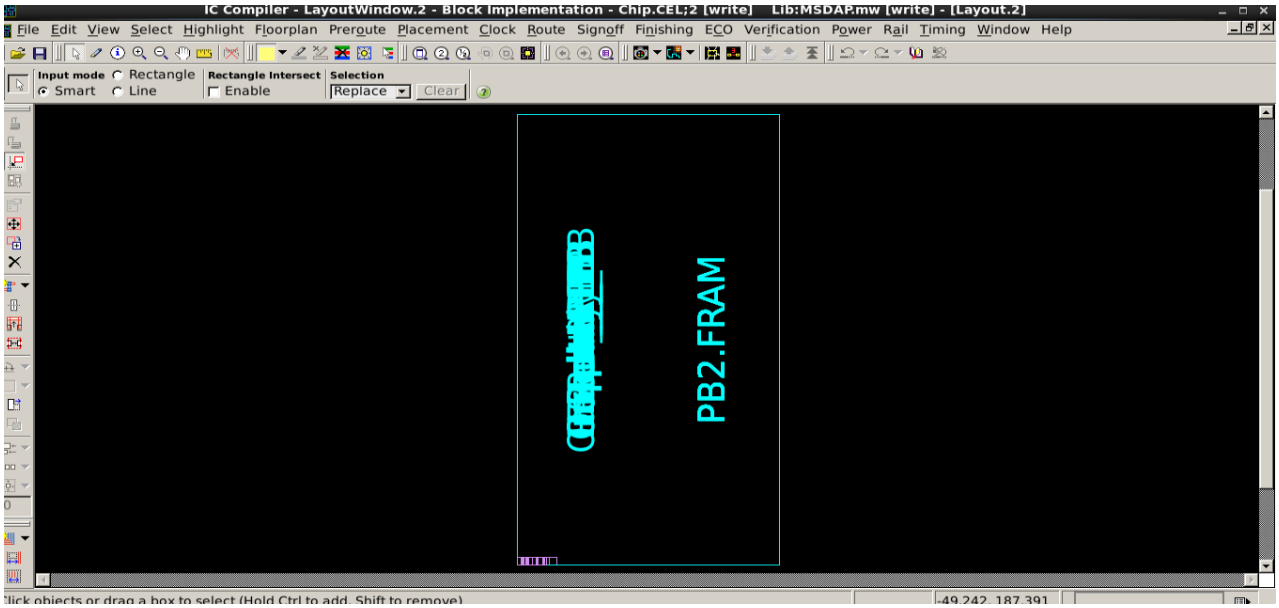
Step 1: Design Entry

The first stage in physical design flow is reading in the netlist and the constraints to your tool of choice. Physical Design converts a gate level netlist into a physical layout.

Gate level netlist: .v file obtained after synthesis.

Standard cell library: This is collection of logic functions like AND, OR etc. Key information from the library such as the location of metal and input/output pins that is required by place and route tools. This standard cell libraries will contain timing information about the function such as cell delay and input pin capacitance which is used to calculated output loads.

Design Constraints: The design constraints are derived from the system specifications. The constraints file contains most designs include clock speeds for each clock in the design.



The layout window shows the layout information.

Step 2: Partitioning

Partitioning is the first step in the physical design flow. Each gate in the design is mapped to a node and a net is mapped as an edge and a graph is constructed. Given a net list of size ‘n’ and area ‘A’, partitioning algorithm such as simulated annealing, Kernighan Lin and F-M algorithms tries to do the bi-partition so that the resulting sizes has area approximately A/2. Partitioning is performed to make floor planning and placement much easier by reducing the number of nets that crosses the partition. This is called Cut-set reduction. Tool does a number of iterations to arrive at global minima and the final configuration is passed to the floor plan and partitioning phase.

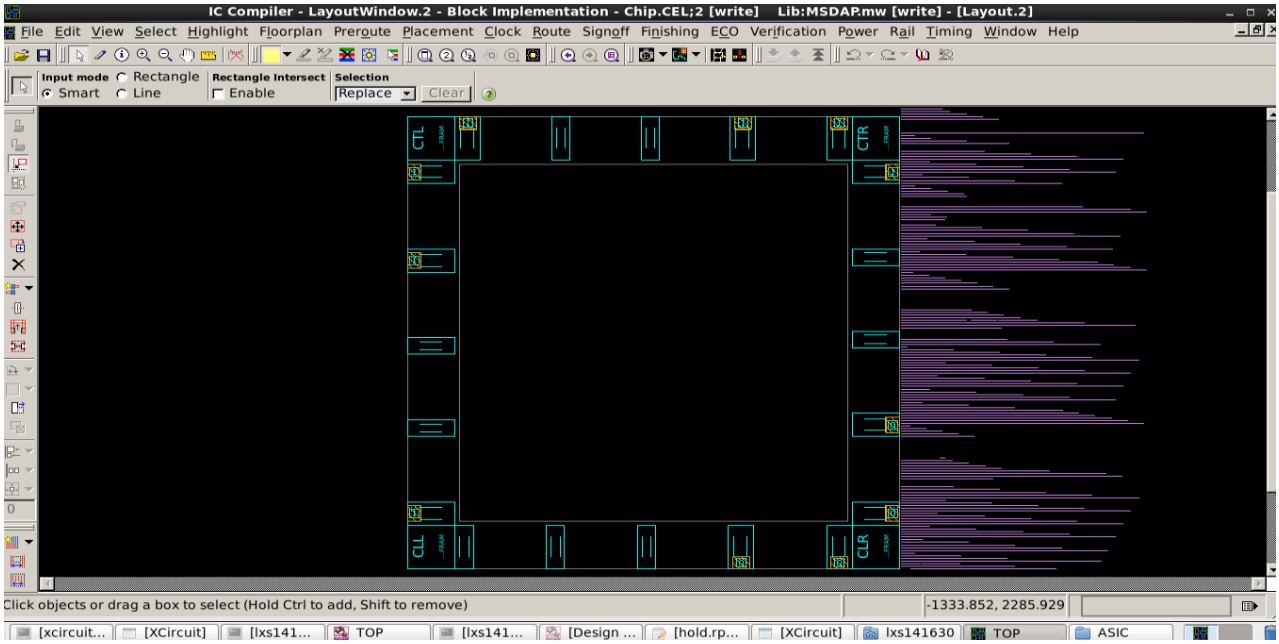
Step 3: Floor planning

After circuit partitioning, floor planning and placement is done iteratively to reduce the local hotspots, critical path delay and area of the chip. There are numerous approach to obtain a global minima for the parameters mentioned above. Floor planning is done using the slicing tree approach and most commonly used approach for placement is simulated annealing, which gives the designer an advantage to choose the cost function based on the requirements.

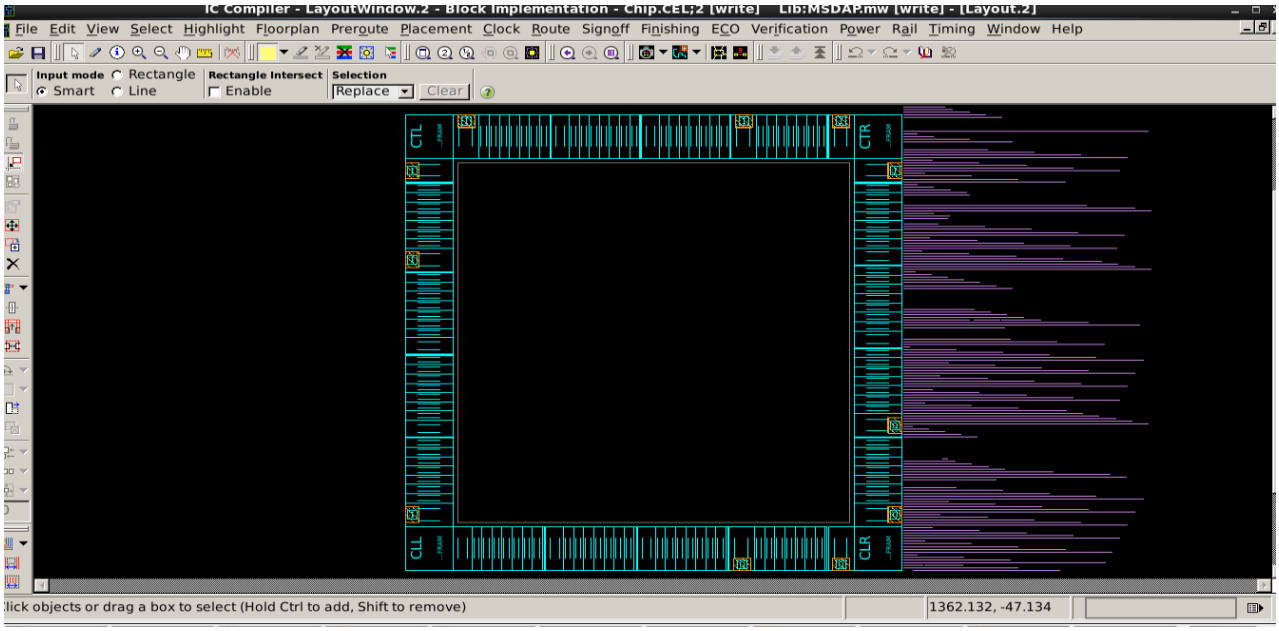
Floor planning control parameters like aspect ratio, core utilization are defined as follows:

Aspect Ratio= Horizontal Routing Resources / Vertical Routing Resources

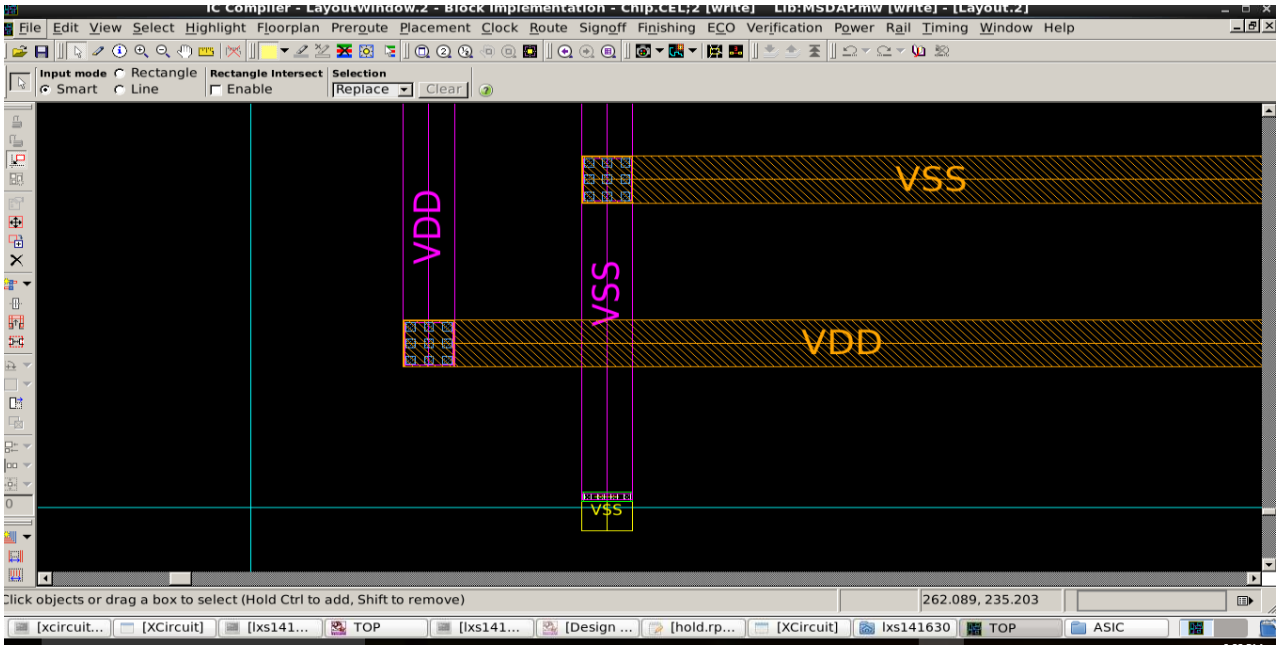
Core Utilization= Standard Cell Area / (Row Area + Channel Area)



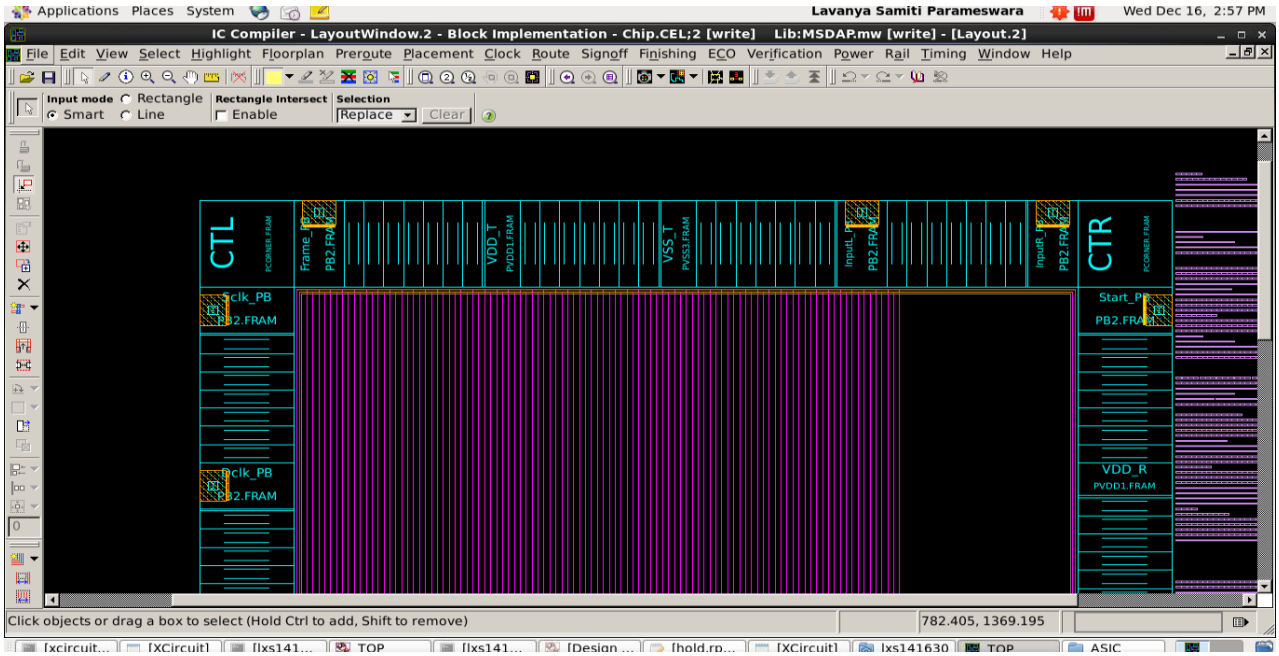
The above picture shows the floor planning for the layout.



The above picture shows the insertion of pad filler.



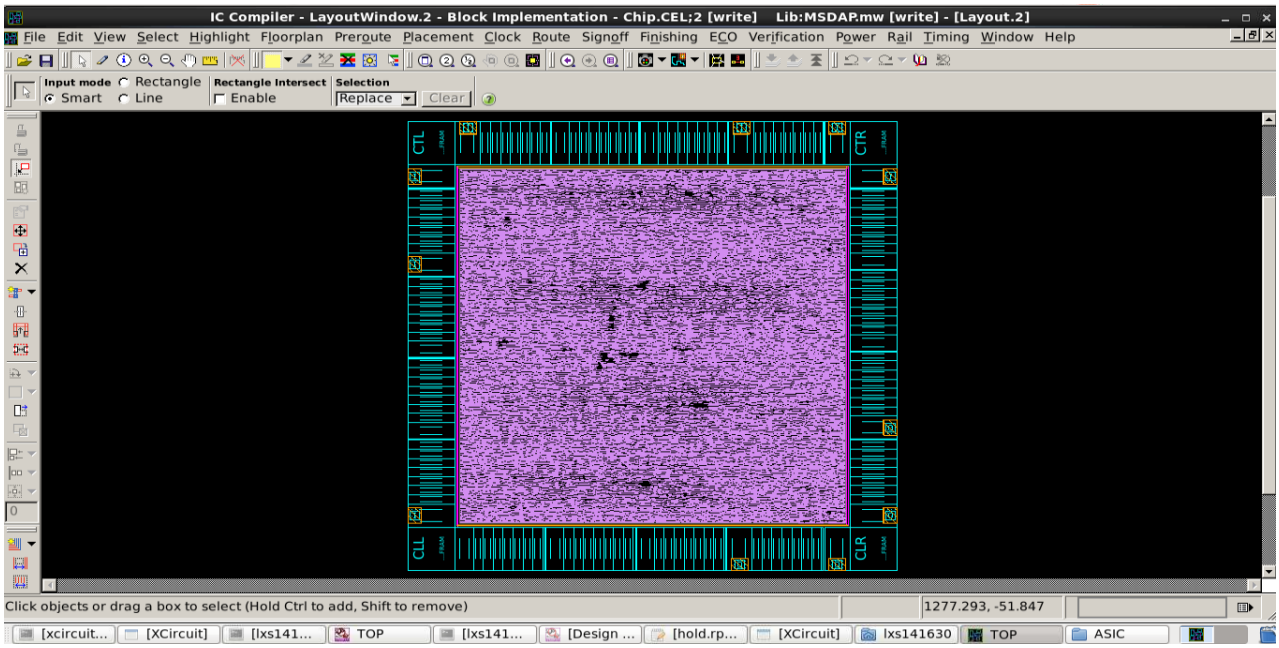
Create VDD/VSS rings.



Create power strap (VDD/VSS).

Step 4: Placement

The process of arranging circuit components on a layout surface. After placement, we can see the accurate estimates of the capacitive loads of each standard cell must drive. The tool places these cells based on the algorithms which it uses internally.



Layout after placement.

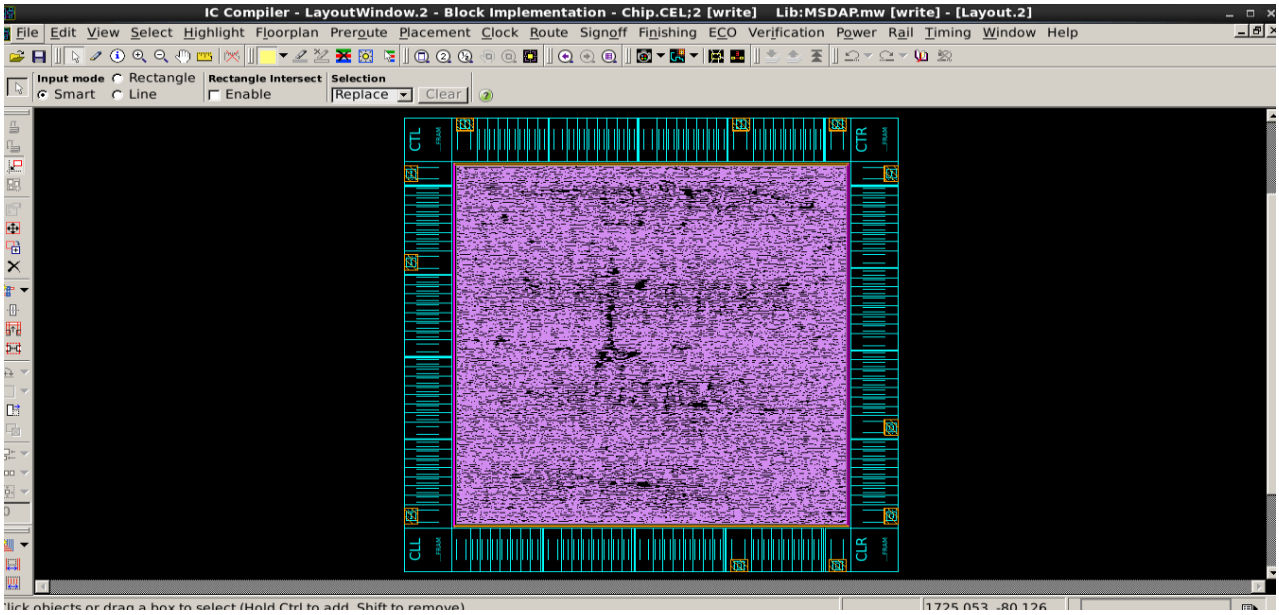
Step 5: Clock Tree Synthesis

Placement and routing goal is always to meet timing constraints provided by the .sdc file. If latency and uncertainty are not set for clock at front end then at backend doing clock tree synthesis (CTS) is not possible.

Cell delay and net delay are stored as look up table.

Cell delay consists of transition, timing arcs and capacitances while net delay is constituted by RCs only. Cell Delays are available in libraries.

The goal of CTS is to minimize skew and insertion delay. Clock is not propagated before CTS.



After CTS optimization.

Step 6: Routing

Routing is a connection of the various standard cells using wires. There are two types of routing.

Global routing : This method first partitions the routing into tiles and then decides tile-to-tile paths for all nets while attempting to optimize some of the objective functions such as circuit timing and the lengths etc.

Detailed routing: This method uses the paths obtained by global routing and assigns tracks and vias for nets.

Physical Verification

Spice file is generated from the layout after no DRC and LVS violations. This spice file is used for testing the chip post layout simulation. The results at this stage should match with the results expected for the design to be perfect functionally. If there is any failure or mismatch, the designer should go back to floor planning and automatic place and route step and see if there can be modifications done to get the functionally correct layout. If this cannot be fixed, then the design has to be changed in the RTL coding step and the steps from there should be repeated until the expected results are obtained. This will be the final testing on the chip before it is sent to Fabrication process. If there are any violations in the timing constraints, then the designer has to go back and see if he can fix the errors in floor planning and placement and continue the process again from that step.

1. Steps for Design Rule Check (DRC)

DRC errors will help the user to find if there are any errors that are against the process technology rules. In IC compiler, to check the DRC errors follow the below steps:

- Choose verification option on the menu panel.
- Select DRC.
- Tick the relevant checks required.

2. Steps for Layout Verses Schematic Check (LVS)

LVS errors will help the designer to know if there any shorts in the design, or any net mismatches occurred during the physical design flow.
In IC compiler, to check the LVS errors follow the below steps:

- Choose verification option on the menu panel
- Select LVS
- Tick the relevant checks required

3. Steps for Antenna Check

Antenna check is an advanced and complex check on every metal layer. It checks the area ratio of all the metal layers in the design and the related capacitances.

- Click on verification menu on the ICC and select the Antenna check to perform the antenna check on the design.

4. Steps for ERC

ERC is electrical rule check that makes sure the circuit is manufactured properly. This is done by checking the relative positions of the cells in the final layout.

- Click on verification menu on the ICC and select the ERC to perform the ERC check on the design.

Challenges in Physical design

- According to the Moore's law, the number of transistors for the same area doubles every year.
- Because of this, the wiring costs increases every time the process is changed. So, a designer has to come up with a new method to have less routing wires and to reduce the total wire length in the design.
- As the number of transistors is increased, the dynamic switching power also increases which has to be handled.
- The IR drop problem also increases as the technology increases which results in more signal integrity issues.
- With all the above issues, speed and accuracy of the chip is at extreme stake.
- Hence, there should be more optimization in floor planning, placement, routing, physical verification and integration.

Optimization

Layout View: For 60% utilization

Std cell utilization: **60.59%** (1215639/(2006250-0)) (Non-fixed + Fixed)

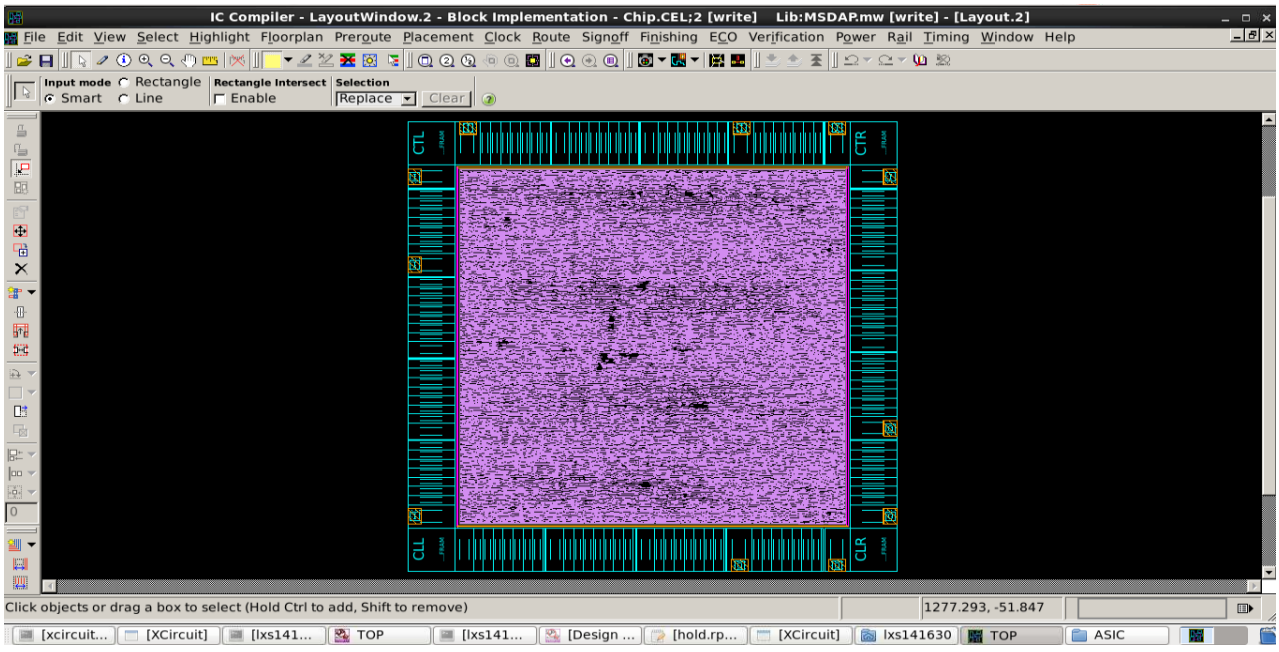
Std cell utilization: 60.59% (1215639/(2006250-0)) (Non-fixed only)

Chip area**:** **2006250** sites, bbox (258.00 258.00 2358.00 2355.20) um

Std cell area: 1215639 sites, (non-fixed:1215639 fixed:0)

77545 cells, (non-fixed:77545 fixed:0)

Lib cell count: **116**



Layout view for cell utilization 60.59%

Layout View: For 70% utilization

Std cell utilization: **70.14%** (1207834/(1722112-0)) (Non-fixed + Fixed)

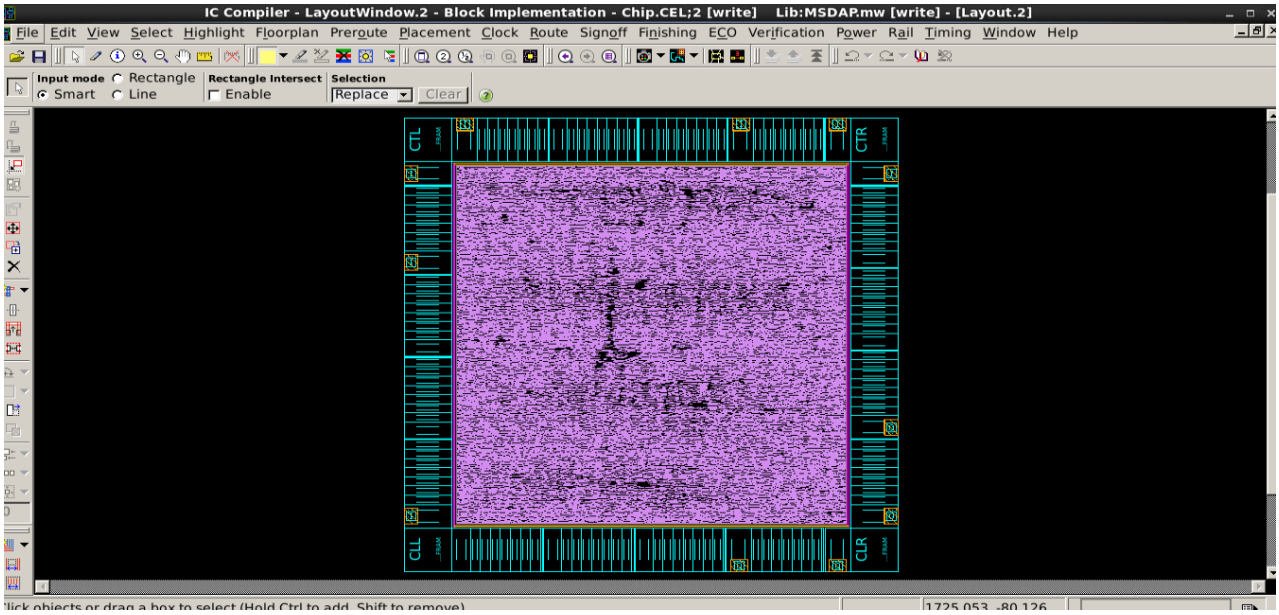
Std cell utilization: 70.14% (1207834/(1722112-0)) (Non-fixed only)

Chip area**:** **1722112** sites, bbox (258.00 258.00 2202.32 2202.32) um

Std cell area: 1207834 sites, (non-fixed:1207834 fixed:0)

77499 cells, (non-fixed:77499 fixed:0)

Lib cell count**:** **126**



Layout view for cell utilization 70.14%

In comparison to 60.59% and 70.14% utilization, the chip area is reduced with the 70.14% cell utilization ratio.

Summary of the results

Std Cell Utilization	**60.59%**	**70.14%**
Chip area	2006250 um^2	1722112 um^2
Std cell area	1215639 um^2	1207834 um^2
Lib cell count	116	126
Total Dynamic Power	3.5728 mW	3.5517 mW
Cell Leakage Power	106.1986 uW	106.1446 uW
Frequency of Sclk	37.2 ns	37.2ns
Slack	MET	MET

Conclusion

- ASIC Design flow of MSDAP is successfully completed.
- RTL Verilog code for MSDAP was written and tested using test bench.
- Timing reports was generated and slack has been met.
- Layout of the MSDAP chip has been implemented using IC compiler.
- After Physical design, the chip area was found to be 1722112 um^2 after optimization.

Appendix: Results

1. Utilization Report

```
****************************************
  Report : Chip Summary
  Design : Chip
  Version: D-2010.03-ICC-SP3
  Date   : Wed Dec 16 15:33:59 2015
****************************************
Std cell utilization: 70.14%  (1207834/(1722112-0))
(Non-fixed + Fixed)
Std cell utilization: 70.14%  (1207834/(1722112-0))
(Non-fixed only)
Chip area:          1722112  sites, bbox (258.00 258.00 2202.32 2202.32) um
Std cell area:      1207834  sites, (non-fixed:1207834 fixed:0)
                    77499    cells, (non-fixed:77499  fixed:0)
Macro cell area:    0        sites
                    0        cells
Placement blockages:  0        sites, (excluding fixed std cells)
                    0        sites, (include fixed std cells & chimney area)
                    0        sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                    0        sites, (routing blockages and signal pre-route)
Lib cell count:     126
Avg. std cell width:  5.65 um
Site array:         unit     (width: 0.56 um, height: 3.92 um, rows: 496)
Physical DB scale:    1000 db_unit = 1 um


****************************************
  Report : pnet options
  Design : Chip
  Version: D-2010.03-ICC-SP3
  Date   : Wed Dec 16 15:33:59 2015
```

****

Layer	Blockage	Min_width	Min_height	Via_additive	Density
METAL1	none	---	---	via additive	---
METAL2	none	---	---	via additive	---
METAL3	none	---	---	via additive	---
METAL4	none	---	---	via additive	---
METAL5	none	---	---	via additive	---
METAL6	none	---	---	via additive	---

****
Sub-Region Utilization
****
Number of regions with placement utilization 0 - 0.125 is 1 (0.01%)
Number of regions with placement utilization 0.125 - 0.25 is 0 (0.00%)
Number of regions with placement utilization 0.25 - 0.375 is 6 (0.06%)
Number of regions with placement utilization 0.375 - 0.5 is 406 (4.06%)
Number of regions with placement utilization 0.5 - 0.625 is 1682 (16.82%)
Number of regions with placement utilization 0.625 - 0.75 is 4420 (44.20%)
Number of regions with placement utilization 0.75 - 0.875 is 3339 (33.39%)
Number of regions with placement utilization 0.875 - 1 is 146 (1.46%)

2. Power report

****
Report : power
-analysis_effort low
Design : Chip
Version: D-2010.03-ICC-SP3
Date : Wed Dec 16 16:04:54 2015
****
Library(s) Used:

SP018EE_V0p5_max (File: /home/eng/x/xxw122030/synopsys/io_max.db)
ss_1v62_125c (File: /home/eng/x/xxw122030/synopsys/ss_1v62_125c.db)

Operating Conditions: ss_1v62_125c Library: ss_1v62_125c
Wire Load Model Mode: top
Global Operating Voltage = 1.62
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 1.000000pf

Time Units = 1ns
Dynamic Power Units = 1mW (derived from V,C,T units)
Leakage Power Units = 1pW
Cell Internal Power = 2.7014 mW (76%)
Net Switching Power = 850.3647 uW (24%)

Total Dynamic Power = 3.5517 mW (100%)

Cell Leakage Power = 106.1446 uW

3. Hold time Report

```
****************************************
Report : timing
        -path full
        -delay min
        -max_paths 20
Design : Chip
Version: D-2010.03-ICC-SP3
Date   : Wed Dec 16 15:50:47 2015
****************************************

 * Some/all delay information is back-annotated.

Operating Conditions: ff_1v98_0c   Library: ff_1v98_0c
            Parasitic source    : LPE
            Parasitic mode      : RealRC
            Extraction mode     : MIN_MAX
            Extraction derating : 125/125/125

  Startpoint: P_MSDAP/main_ctrl/next_state_reg[0]
            (falling edge-triggered flip-flop clocked by Dclk)
  Endpoint: P_MSDAP/main_ctrl/pr_state_reg[0]
           (falling edge-triggered flip-flop clocked by Dclk)
  Path Group: Dclk
  Path Type: min

  Point                                    Incr       Path
  -----------------------------------------------------------------------
  clock Dclk (fall edge)                 651.00     651.00
  clock network delay (propagated)         2.40     653.40
  P_MSDAP/main_ctrl/next_state_reg[0]/CKN (DFFNSRHX1M)
                                         0.00     653.40 f
  P_MSDAP/main_ctrl/next_state_reg[0]/Q (DFFNSRHX1M)      0.73     654.12 r
  P_MSDAP/main_ctrl/pr_state_reg[0]/D (DFFNSRHX1M)        0.00 *   654.12 r
  data arrival time                                      654.12
```

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/main_ctrl/pr_state_reg[0]/CKN (DFFNSRHX1M) 0.00 653.45 f
library hold time 0.21 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.12
---
slack (MET) 0.46

Startpoint: P_MSDAP/main_ctrl/coeff_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/main_ctrl/coeff_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point Incr Path
---
clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/main_ctrl/coeff_count_reg[0]/CKN (DFFNHX2M) 0.00 653.45 f
P_MSDAP/main_ctrl/coeff_count_reg[0]/Q (DFFNHX2M) 0.57 654.02 f
P_MSDAP/main_ctrl/U77/Y (OAI22X1M) 0.22 * 654.24 r
P_MSDAP/main_ctrl/coeff_count_reg[0]/D (DFFNHX2M) 0.00 * 654.24 r
data arrival time 654.24

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/main_ctrl/coeff_count_reg[0]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.25 653.70
data required time 653.70
---
data required time 653.70
data arrival time -654.24
---
slack (MET) 0.54

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)

Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.33	653.33
P_MSDAP/PISO_L/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.33 f
P_MSDAP/PISO_L/piso_reg_reg[3]/Q (DFFNHX2M)	0.59	653.92 r
P_MSDAP/PISO_L/U71/Y (AO22X2M)	0.19 *	654.11 r
P_MSDAP/PISO_L/piso_reg_reg[3]/D (DFFNHX2M)	0.00 *	654.11 r
data arrival time		654.11
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.33	653.33
P_MSDAP/PISO_L/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.33 f
library hold time	0.24	653.57
data required time		653.57
data required time		653.57
data arrival time		-654.11
slack (MET)		0.55

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[37]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[37]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[37]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[37]/Q (DFFNHX2M)	0.60	654.05 r
P_MSDAP/PISO_R/U65/Y (AO22X2M)	0.19 *	654.24 r
P_MSDAP/PISO_R/piso_reg_reg[37]/D (DFFNHX2M)	0.00 *	654.24 r
data arrival time		654.24
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[37]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.67

data required time 653.67
---
data required time 653.67
data arrival time -654.24
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[3]/Q (DFFNHX2M)	0.59	654.04 r
P_MSDAP/PISO_R/U71/Y (AO22X2M)	0.20 *	654.24 r
P_MSDAP/PISO_R/piso_reg_reg[3]/D (DFFNHX2M)	0.00 *	654.24 r
data arrival time		654.24
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.67
data required time		653.67

---
data required time 653.67
data arrival time -654.24
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[0]/CKN (DFFNHX2M) 0.00 653.45 f
P_MSDAP/PISO_R/piso_reg_reg[0]/Q (DFFNHX2M) 0.60 654.05 r
P_MSDAP/PISO_R/U33/Y (AO22X2M) 0.19 * 654.24 r
P_MSDAP/PISO_R/piso_reg_reg[0]/D (DFFNHX2M) 0.00 * 654.24 r
data arrival time 654.24

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[0]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.24
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[29]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[29]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point Incr Path
---
clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[29]/CKN (DFFNHX2M) 0.00 653.32 f
P_MSDAP/PISO_L/piso_reg_reg[29]/Q (DFFNHX2M) 0.61 653.93 r
P_MSDAP/PISO_L/U57/Y (AO22X2M) 0.19 * 654.12 r
P_MSDAP/PISO_L/piso_reg_reg[29]/D (DFFNHX2M) 0.00 * 654.12 r
data arrival time 654.12

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[29]/CKN (DFFNHX2M) 0.00 653.32 f
library hold time 0.23 653.55
data required time 653.55
---
data required time 653.55
data arrival time -654.12
---

slack (MET) 0.57

Startpoint: P_MSDAP/main_ctrl/data_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/main_ctrl/data_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.47	653.47
P_MSDAP/main_ctrl/data_count_reg[0]/CKN (DFFNHX2M)	0.00	653.47 f
P_MSDAP/main_ctrl/data_count_reg[0]/Q (DFFNHX2M)	0.55	654.03 f
P_MSDAP/main_ctrl/r100/A[0] (MSDAP_controller_DW01_inc_2)	0.00	654.03 f
P_MSDAP/main_ctrl/r100/U1/Y (INVX2M)	0.06 *	654.09 r
P_MSDAP/main_ctrl/r100/SUM[0] (MSDAP_controller_DW01_inc_2)	0.00	654.09 r
P_MSDAP/main_ctrl/U120/Y (OAI2BB2X1M)	0.21 *	654.30 r
P_MSDAP/main_ctrl/data_count_reg[0]/D (DFFNHX2M)	0.00 *	654.30 r
data arrival time		654.30
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.47	653.47
P_MSDAP/main_ctrl/data_count_reg[0]/CKN (DFFNHX2M)	0.00	653.47 f
library hold time	0.25	653.73
data required time		653.73
data required time		653.73
data arrival time		-654.30
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[13]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[13]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.33 653.33
P_MSDAP/PISO_L/piso_reg_reg[13]/CKN (DFFNHX2M) 0.00 653.33 f
P_MSDAP/PISO_L/piso_reg_reg[13]/Q (DFFNHX2M) 0.62 653.95 r
P_MSDAP/PISO_L/U41/Y (AO22X2M) 0.19 * 654.14 r
P_MSDAP/PISO_L/piso_reg_reg[13]/D (DFFNHX2M) 0.00 * 654.14 r
data arrival time 654.14

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.33 653.33
P_MSDAP/PISO_L/piso_reg_reg[13]/CKN (DFFNHX2M) 0.00 653.33 f
library hold time 0.24 653.57
data required time 653.57
---
data required time 653.57
data arrival time -654.14
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[11]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[11]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point Incr Path
---
clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[11]/CKN (DFFNHX2M) 0.00 653.45 f
P_MSDAP/PISO_R/piso_reg_reg[11]/Q (DFFNHX2M) 0.60 654.05 r
P_MSDAP/PISO_R/U39/Y (AO22X2M) 0.20 * 654.25 r
P_MSDAP/PISO_R/piso_reg_reg[11]/D (DFFNHX2M) 0.00 * 654.25 r
data arrival time 654.25

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[11]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.25
---

slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.33	653.33
P_MSDAP/PISO_L/piso_reg_reg[8]/CKN (DFFNHX2M)	0.00	653.33 f
P_MSDAP/PISO_L/piso_reg_reg[8]/Q (DFFNHX2M)	0.62	653.95 r
P_MSDAP/PISO_L/U36/Y (AO22X2M)	0.19 *	654.14 r
P_MSDAP/PISO_L/piso_reg_reg[8]/D (DFFNHX2M)	0.00 *	654.14 r
data arrival time		654.14
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.33	653.33
P_MSDAP/PISO_L/piso_reg_reg[8]/CKN (DFFNHX2M)	0.00	653.33 f
library hold time	0.24	653.57
data required time		653.57
data required time		653.57
data arrival time		-654.14
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[37]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[37]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.33	653.33
P_MSDAP/PISO_L/piso_reg_reg[37]/CKN (DFFNHX2M)	0.00	653.33 f
P_MSDAP/PISO_L/piso_reg_reg[37]/Q (DFFNHX2M)	0.62	653.95 r
P_MSDAP/PISO_L/U65/Y (AO22X2M)	0.20 *	654.14 r

P_MSDAP/PISO_L/piso_reg_reg[37]/D (DFFNHX2M) 0.00 * 654.14 r
data arrival time 654.14

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.33 653.33
P_MSDAP/PISO_L/piso_reg_reg[37]/CKN (DFFNHX2M) 0.00 653.33 f
library hold time 0.24 653.57
data required time 653.57
---
data required time 653.57
data arrival time -654.14
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[16]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[16]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point Incr Path
---
clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[16]/CKN (DFFNHX2M) 0.00 653.32 f
P_MSDAP/PISO_L/piso_reg_reg[16]/Q (DFFNHX2M) 0.61 653.93 r
P_MSDAP/PISO_L/U44/Y (AO22X2M) 0.20 * 654.12 r
P_MSDAP/PISO_L/piso_reg_reg[16]/D (DFFNHX2M) 0.00 * 654.12 r
data arrival time 654.12

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[16]/CKN (DFFNHX2M) 0.00 653.32 f
library hold time 0.23 653.55
data required time 653.55
---
data required time 653.55
data arrival time -654.12
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[22]
(falling edge-triggered flip-flop clocked by Dclk)

Endpoint: P_MSDAP/PISO_L/piso_reg_reg[22]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[22]/CKN (DFFNHX2M)	0.00	653.32 f
P_MSDAP/PISO_L/piso_reg_reg[22]/Q (DFFNHX2M)	0.61	653.93 r
P_MSDAP/PISO_L/U50/Y (AO22X2M)	0.19 *	654.13 r
P_MSDAP/PISO_L/piso_reg_reg[22]/D (DFFNHX2M)	0.00 *	654.13 r
data arrival time		654.13
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[22]/CKN (DFFNHX2M)	0.00	653.32 f
library hold time	0.23	653.55
data required time		653.55
data required time		653.55
data arrival time		-654.13
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[14]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[14]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[14]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[14]/Q (DFFNHX2M)	0.60	654.06 r
P_MSDAP/PISO_R/U42/Y (AO22X2M)	0.19 *	654.25 r
P_MSDAP/PISO_R/piso_reg_reg[14]/D (DFFNHX2M)	0.00 *	654.25 r
data arrival time		654.25
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45

P_MSDAP/PISO_R/piso_reg_reg[14]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.25
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[8]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[8]/Q (DFFNHX2M)	0.60	654.06 r
P_MSDAP/PISO_R/U36/Y (AO22X2M)	0.19 *	654.25 r
P_MSDAP/PISO_R/piso_reg_reg[8]/D (DFFNHX2M)	0.00 *	654.25 r
data arrival time		654.25
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[8]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.68
data required time		653.68
data required time		653.68
data arrival time		-654.25
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[21]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[21]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

```
Point                                                   Incr       Path
-----------------------------------------------------------------------------
clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[21]/CKN (DFFNHX2M)          0.00     653.32 f
P_MSDAP/PISO_L/piso_reg_reg[21]/Q (DFFNHX2M)            0.61     653.93 r
P_MSDAP/PISO_L/U49/Y (AO22X2M)                          0.19 *   654.13 r
P_MSDAP/PISO_L/piso_reg_reg[21]/D (DFFNHX2M)            0.00 *   654.13 r
data arrival time                                             654.13

clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[21]/CKN (DFFNHX2M)          0.00     653.32 f
library hold time                                       0.23     653.55
data required time                                               653.55
-----------------------------------------------------------------------------
data required time                                               653.55
data arrival time                                               -654.13
-----------------------------------------------------------------------------
slack (MET)                                                      0.57


Startpoint: P_MSDAP/PISO_L/piso_reg_reg[19]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[19]
          (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                                   Incr       Path
-----------------------------------------------------------------------------
clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[19]/CKN (DFFNHX2M)          0.00     653.32 f
P_MSDAP/PISO_L/piso_reg_reg[19]/Q (DFFNHX2M)            0.61     653.93 r
P_MSDAP/PISO_L/U47/Y (AO22X2M)                          0.19 *   654.13 r
P_MSDAP/PISO_L/piso_reg_reg[19]/D (DFFNHX2M)            0.00 *   654.13 r
data arrival time                                             654.13

clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[19]/CKN (DFFNHX2M)          0.00     653.32 f
library hold time                                       0.23     653.55
data required time                                               653.55
-----------------------------------------------------------------------------
data required time                                               653.55
```

```
data arrival time                                  -654.13
-------------------------------------------------------------------------
slack (MET)                                         0.57


Startpoint: P_MSDAP/PISO_L/piso_reg_reg[25]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[25]
         (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                      Incr       Path
-------------------------------------------------------------------------
clock Dclk (fall edge)                   651.00     651.00
clock network delay (propagated)           2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[25]/CKN (DFFNHX2M)        0.00     653.32 f
P_MSDAP/PISO_L/piso_reg_reg[25]/Q (DFFNHX2M)          0.61     653.94 r
P_MSDAP/PISO_L/U53/Y (AO22X2M)                        0.19 *   654.13 r
P_MSDAP/PISO_L/piso_reg_reg[25]/D (DFFNHX2M)          0.00 *   654.13 r
data arrival time                                  654.13

clock Dclk (fall edge)                   651.00     651.00
clock network delay (propagated)           2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[25]/CKN (DFFNHX2M)        0.00     653.32 f
library hold time                        0.23     653.55
data required time                                 653.55
-------------------------------------------------------------------------
data required time                                 653.55
data arrival time                                 -654.13
-------------------------------------------------------------------------
slack (MET)                                        0.57


Startpoint: P_MSDAP/PISO_R/piso_reg_reg[10]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[10]
         (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                      Incr       Path
-------------------------------------------------------------------------
clock Dclk (fall edge)                   651.00     651.00
clock network delay (propagated)           2.45     653.45
P_MSDAP/PISO_R/piso_reg_reg[10]/CKN (DFFNHX2M)        0.00     653.45 f
```

P_MSDAP/PISO_R/piso_reg_reg[10]/Q (DFFNHX2M) 0.60 654.06 r
P_MSDAP/PISO_R/U38/Y (AO22X2M) 0.19 * 654.25 r
P_MSDAP/PISO_R/piso_reg_reg[10]/D (DFFNHX2M) 0.00 * 654.25 r
data arrival time 654.25

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[10]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.25
---
slack (MET) 0.58

Startpoint: P_MSDAP/SIPO_uut/bit_count_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/bit_count_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/bit_count_reg[0]/CKN (DFFNSRHX1M) 0.00 16.33 f
P_MSDAP/SIPO_uut/bit_count_reg[0]/Q (DFFNSRHX1M) 0.83 17.16 f
P_MSDAP/SIPO_uut/U105/Y (NAND2X2M) 0.15 * 17.31 r
P_MSDAP/SIPO_uut/bit_count_reg[0]/D (DFFNSRHX1M) 0.00 * 17.31 r
data arrival time 17.31

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/bit_count_reg[0]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.31
---
slack (MET) 0.82

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[4]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[4]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[4]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_L_reg[4]/Q (DFFNSRHX1M)	0.80	17.13 r
P_MSDAP/SIPO_uut/U110/Y (OAI2BB2X1M)	0.22 *	17.35 r
P_MSDAP/SIPO_uut/data_L_reg[4]/D (DFFNSRHX1M)	0.00 *	17.35 r
data arrival time		17.35
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[4]/CKN (DFFNSRHX1M)	0.00	16.33 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.35
slack (MET)		0.85

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[0]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[0]/Q (DFFNSRHX1M)	0.79	17.13 r
P_MSDAP/SIPO_uut/U85/Y (OAI2BB2X1M)	0.23 *	17.36 r
P_MSDAP/SIPO_uut/data_L_reg[0]/D (DFFNSRHX1M)	0.00 *	17.36 r
data arrival time		17.36

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[0]/CKN (DFFNSRHX1M) 0.00 16.34 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.36
---
slack (MET) 0.86

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[9]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[9]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[9]/CKN (DFFNSRHX1M) 0.00 16.34 f
P_MSDAP/SIPO_uut/data_L_reg[9]/Q (DFFNSRHX1M) 0.80 17.14 r
P_MSDAP/SIPO_uut/U94/Y (OAI2BB2X1M) 0.23 * 17.37 r
P_MSDAP/SIPO_uut/data_L_reg[9]/D (DFFNSRHX1M) 0.00 * 17.37 r
data arrival time 17.37

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[9]/CKN (DFFNSRHX1M) 0.00 16.34 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.37
---
slack (MET) 0.86

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[5]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[5]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk

Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[5]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_L_reg[5]/Q (DFFNSRHX1M)	0.80	17.14 r
P_MSDAP/SIPO_uut/U119/Y (OAI2BB2X1M)	0.23 *	17.37 r
P_MSDAP/SIPO_uut/data_L_reg[5]/D (DFFNSRHX1M)	0.00 *	17.37 r
data arrival time		17.37
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[5]/CKN (DFFNSRHX1M)	0.00	16.33 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.37
slack (MET)		0.87

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[1]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[1]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[1]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[1]/Q (DFFNSRHX1M)	0.80	17.13 r
P_MSDAP/SIPO_uut/U92/Y (OAI2BB2X1M)	0.24 *	17.37 r
P_MSDAP/SIPO_uut/data_L_reg[1]/D (DFFNSRHX1M)	0.00 *	17.37 r
data arrival time		17.37
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[1]/CKN (DFFNSRHX1M)	0.00	16.34 f
library hold time	0.17	16.50
data required time		16.50

```
---------------------------------------------------------------------------
data required time                                              16.50
data arrival time                                              -17.37
---------------------------------------------------------------------------
slack (MET)                                                     0.87


Startpoint: P_MSDAP/SIPO_uut/data_L_reg[11]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[11]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                          Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                        15.00      15.00
clock network delay (propagated)               1.34      16.34
P_MSDAP/SIPO_uut/data_L_reg[11]/CKN (DFFNSRHX1M)       0.00      16.34 f
P_MSDAP/SIPO_uut/data_L_reg[11]/Q (DFFNSRHX1M)         0.80      17.14 r
P_MSDAP/SIPO_uut/U120/Y (OAI2BB2X1M)                   0.24 *    17.38 r
P_MSDAP/SIPO_uut/data_L_reg[11]/D (DFFNSRHX1M)         0.00 *    17.38 r
data arrival time                                              17.38

clock Sclk (fall edge)                        15.00      15.00
clock network delay (propagated)               1.34      16.34
P_MSDAP/SIPO_uut/data_L_reg[11]/CKN (DFFNSRHX1M)       0.00      16.34 f
library hold time                              0.17      16.50
data required time                                             16.50
---------------------------------------------------------------------------
data required time                                             16.50
data arrival time                                             -17.38
---------------------------------------------------------------------------
slack (MET)                                                    0.87


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[9]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[9]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                          Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                        15.00      15.00
```

clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_R_reg[9]/CKN (DFFNSRHX1M) 0.00 16.33 f
P_MSDAP/SIPO_uut/data_R_reg[9]/Q (DFFNSRHX1M) 0.81 17.15 r
P_MSDAP/SIPO_uut/U91/Y (OAI2BB2X1M) 0.23 * 17.37 r
P_MSDAP/SIPO_uut/data_R_reg[9]/D (DFFNSRHX1M) 0.00 * 17.37 r
data arrival time 17.37

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_R_reg[9]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.37
---
slack (MET) 0.88

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[13]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[13]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_L_reg[13]/CKN (DFFNSRHX1M) 0.00 16.33 f
P_MSDAP/SIPO_uut/data_L_reg[13]/Q (DFFNSRHX1M) 0.82 17.15 r
P_MSDAP/SIPO_uut/U121/Y (OAI2BB2X1M) 0.23 * 17.38 r
P_MSDAP/SIPO_uut/data_L_reg[13]/D (DFFNSRHX1M) 0.00 * 17.38 r
data arrival time 17.38

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_L_reg[13]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.38
---
slack (MET) 0.88

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[7]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[7]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[7]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[7]/Q (DFFNSRHX1M)	0.81	17.15 r
P_MSDAP/SIPO_uut/U93/Y (OAI2BB2X1M)	0.23 *	17.38 r
P_MSDAP/SIPO_uut/data_L_reg[7]/D (DFFNSRHX1M)	0.00 *	17.38 r
data arrival time		17.38
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[7]/CKN (DFFNSRHX1M)	0.00	16.34 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.38
slack (MET)		0.88

Startpoint: P_MSDAP/SIPO_uut/data_R_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[0]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_R_reg[0]/Q (DFFNSRHX1M)	0.82	17.15 r
P_MSDAP/SIPO_uut/U81/Y (OAI2BB2X1M)	0.23 *	17.38 r
P_MSDAP/SIPO_uut/data_R_reg[0]/D (DFFNSRHX1M)	0.00 *	17.38 r

```
data arrival time                                                  17.38

clock Sclk (fall edge)                                 15.00      15.00
clock network delay (propagated)                        1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[0]/CKN (DFFNSRHX1M)         0.00      16.33 f
library hold time                                      0.17       16.50
data required time                                                16.50
-----------------------------------------------------------------------------
data required time                                                16.50
data arrival time                                                -17.38
-----------------------------------------------------------------------------
slack (MET)                                                       0.88


Startpoint: P_MSDAP/SIPO_uut/data_L_reg[3]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[3]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                                  Incr       Path
-----------------------------------------------------------------------------
clock Sclk (fall edge)                                 15.00      15.00
clock network delay (propagated)                        1.33      16.33
P_MSDAP/SIPO_uut/data_L_reg[3]/CKN (DFFNSRHX1M)         0.00      16.33 f
P_MSDAP/SIPO_uut/data_L_reg[3]/Q (DFFNSRHX1M)           0.81      17.14 r
P_MSDAP/SIPO_uut/U118/Y (OAI2BB2X1M)                  0.24 *    17.38 r
P_MSDAP/SIPO_uut/data_L_reg[3]/D (DFFNSRHX1M)         0.00 *    17.38 r
data arrival time                                                 17.38

clock Sclk (fall edge)                                 15.00      15.00
clock network delay (propagated)                        1.33      16.33
P_MSDAP/SIPO_uut/data_L_reg[3]/CKN (DFFNSRHX1M)         0.00      16.33 f
library hold time                                      0.17       16.50
data required time                                                16.50
-----------------------------------------------------------------------------
data required time                                                16.50
data arrival time                                                -17.38
-----------------------------------------------------------------------------
slack (MET)                                                       0.88


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[5]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[5]
```

(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[5]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_R_reg[5]/Q (DFFNSRHX1M)	0.81	17.14 r
P_MSDAP/SIPO_uut/U115/Y (OAI2BB2X1M)	0.24 *	17.39 r
P_MSDAP/SIPO_uut/data_R_reg[5]/D (DFFNSRHX1M)	0.00 *	17.39 r
data arrival time		17.39
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[5]/CKN (DFFNSRHX1M)	0.00	16.33 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.39
slack (MET)		0.89

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[2]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[2]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[2]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_L_reg[2]/Q (DFFNSRHX1M)	0.82	17.16 r
P_MSDAP/SIPO_uut/U86/Y (OAI2BB2X1M)	0.23 *	17.39 r
P_MSDAP/SIPO_uut/data_L_reg[2]/D (DFFNSRHX1M)	0.00 *	17.39 r
data arrival time		17.39
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[2]/CKN (DFFNSRHX1M)	0.00	16.33 f

library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.39
---
slack (MET) 0.89

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[6]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[6]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path	
clock Sclk (fall edge)	15.00	15.00	
clock network delay (propagated)	1.34	16.34	
P_MSDAP/SIPO_uut/data_L_reg[6]/CKN (DFFNSRHX1M)	0.00	16.34	f
P_MSDAP/SIPO_uut/data_L_reg[6]/Q (DFFNSRHX1M)	0.83	17.16	r
P_MSDAP/SIPO_uut/U111/Y (OAI2BB2X1M)	0.23 *	17.40	r
P_MSDAP/SIPO_uut/data_L_reg[6]/D (DFFNSRHX1M)	0.00 *	17.40	r
data arrival time		17.40	

clock Sclk (fall edge)	15.00	15.00	
clock network delay (propagated)	1.34	16.34	
P_MSDAP/SIPO_uut/data_L_reg[6]/CKN (DFFNSRHX1M)	0.00	16.34	f
library hold time	0.17	16.50	
data required time		16.50	
data required time		16.50	
data arrival time		-17.40	
slack (MET)		0.89	

Startpoint: P_MSDAP/SIPO_uut/bit_count_reg[3]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/bit_count_reg[3]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path

```
---------------------------------------------------------------------------
clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.33      16.33
P_MSDAP/SIPO_uut/bit_count_reg[3]/CKN (DFFNSRHX1M)      0.00      16.33 f
P_MSDAP/SIPO_uut/bit_count_reg[3]/Q (DFFNSRHX1M)        0.77      17.10 r
P_MSDAP/SIPO_uut/U75/Y (XOR2X2M)                 0.19 *    17.29 f
P_MSDAP/SIPO_uut/U51/Y (NAND2X2M)                0.10 *    17.39 r
P_MSDAP/SIPO_uut/bit_count_reg[3]/D (DFFNSRHX1M)        0.00 *    17.39 r
data arrival time                                17.39

clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.33      16.33
P_MSDAP/SIPO_uut/bit_count_reg[3]/CKN (DFFNSRHX1M)      0.00      16.33 f
library hold time                               0.17      16.50
data required time                               16.50
---------------------------------------------------------------------------
data required time                               16.50
data arrival time                               -17.39
---------------------------------------------------------------------------
slack (MET)                                      0.90


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[15]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[15]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                           Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[15]/CKN (DFFNSRHX1M)        0.00      16.33 f
P_MSDAP/SIPO_uut/data_R_reg[15]/Q (DFFNSRHX1M)          0.83      17.16 r
P_MSDAP/SIPO_uut/U95/Y (OAI2BB2X1M)              0.24 *    17.40 r
P_MSDAP/SIPO_uut/data_R_reg[15]/D (DFFNSRHX1M)          0.00 *    17.40 r
data arrival time                                17.40

clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[15]/CKN (DFFNSRHX1M)        0.00      16.33 f
library hold time                               0.17      16.50
data required time                               16.50
---------------------------------------------------------------------------
data required time                               16.50
```

```
data arrival time                                  -17.40
-------------------------------------------------------------------------
slack (MET)                                         0.90


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[13]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[13]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                       Incr       Path
-------------------------------------------------------------------------
clock Sclk (fall edge)                     15.00      15.00
clock network delay (propagated)            1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[13]/CKN (DFFNSRHX1M)       0.00      16.33 f
P_MSDAP/SIPO_uut/data_R_reg[13]/Q (DFFNSRHX1M)         0.84      17.17 r
P_MSDAP/SIPO_uut/U117/Y (OAI2BB2X1M)                   0.23 *    17.41 r
P_MSDAP/SIPO_uut/data_R_reg[13]/D (DFFNSRHX1M)         0.00 *    17.41 r
data arrival time                                  17.41

clock Sclk (fall edge)                     15.00      15.00
clock network delay (propagated)            1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[13]/CKN (DFFNSRHX1M)       0.00      16.33 f
library hold time                          0.17       16.50
data required time                                    16.50
-------------------------------------------------------------------------
data required time                                    16.50
data arrival time                                  -17.41
-------------------------------------------------------------------------
slack (MET)                                         0.91


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[1]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[1]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                       Incr       Path
-------------------------------------------------------------------------
clock Sclk (fall edge)                     15.00      15.00
clock network delay (propagated)            1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[1]/CKN (DFFNSRHX1M)        0.00      16.33 f
```

P_MSDAP/SIPO_uut/data_R_reg[1]/Q (DFFNSRHX1M) 0.84 17.17 r
P_MSDAP/SIPO_uut/U89/Y (OAI2BB2X1M) 0.24 * 17.41 r
P_MSDAP/SIPO_uut/data_R_reg[1]/D (DFFNSRHX1M) 0.00 * 17.41 r
data arrival time 17.41

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_R_reg[1]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.41
---
slack (MET) 0.91

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[12]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[12]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[12]/CKN (DFFNSRHX1M) 0.00 16.34 f
P_MSDAP/SIPO_uut/data_L_reg[12]/Q (DFFNSRHX1M) 0.84 17.18 r
P_MSDAP/SIPO_uut/U112/Y (OAI2BB2X1M) 0.24 * 17.41 r
P_MSDAP/SIPO_uut/data_L_reg[12]/D (DFFNSRHX1M) 0.00 * 17.41 r
data arrival time 17.41

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[12]/CKN (DFFNSRHX1M) 0.00 16.34 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.41
---
slack (MET) 0.91

4. <u>Setup time Report</u>

```
****************************************
Report : timing
        -path full
        -delay min
        -max_paths 20
Design : Chip
Version: D-2010.03-ICC-SP3
Date   : Wed Dec 16 15:50:47 2015
****************************************

 * Some/all delay information is back-annotated.

Operating Conditions: ff_1v98_0c   Library: ff_1v98_0c
            Parasitic source    : LPE
            Parasitic mode      : RealRC
            Extraction mode     : MIN_MAX
            Extraction derating : 125/125/125

  Startpoint: P_MSDAP/main_ctrl/next_state_reg[0]
            (falling edge-triggered flip-flop clocked by Dclk)
  Endpoint: P_MSDAP/main_ctrl/pr_state_reg[0]
           (falling edge-triggered flip-flop clocked by Dclk)
  Path Group: Dclk
  Path Type: min

  Point                                    Incr       Path
  ---------------------------------------------------------------------------
  clock Dclk (fall edge)                 651.00     651.00
  clock network delay (propagated)         2.40     653.40
  P_MSDAP/main_ctrl/next_state_reg[0]/CKN (DFFNSRHX1M)
                                         0.00     653.40 f
  P_MSDAP/main_ctrl/next_state_reg[0]/Q (DFFNSRHX1M)      0.73     654.12 r
  P_MSDAP/main_ctrl/pr_state_reg[0]/D (DFFNSRHX1M)        0.00 *   654.12 r
  data arrival time                                      654.12

  clock Dclk (fall edge)                 651.00     651.00
  clock network delay (propagated)         2.45     653.45
  P_MSDAP/main_ctrl/pr_state_reg[0]/CKN (DFFNSRHX1M)      0.00     653.45 f
  library hold time                      0.21     653.67
  data required time                             653.67
  ---------------------------------------------------------------------------
  data required time                             653.67
  data arrival time                             -654.12
  ---------------------------------------------------------------------------
```

slack (MET) 0.46

Startpoint: P_MSDAP/main_ctrl/coeff_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/main_ctrl/coeff_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/main_ctrl/coeff_count_reg[0]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/main_ctrl/coeff_count_reg[0]/Q (DFFNHX2M)	0.57	654.02 f
P_MSDAP/main_ctrl/U77/Y (OAI22X1M)	0.22 *	654.24 r
P_MSDAP/main_ctrl/coeff_count_reg[0]/D (DFFNHX2M)	0.00 *	654.24 r
data arrival time		654.24
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/main_ctrl/coeff_count_reg[0]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.25	653.70
data required time		653.70
data required time		653.70
data arrival time		-654.24
slack (MET)		0.54

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.33	653.33
P_MSDAP/PISO_L/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.33 f
P_MSDAP/PISO_L/piso_reg_reg[3]/Q (DFFNHX2M)	0.59	653.92 r
P_MSDAP/PISO_L/U71/Y (AO22X2M)	0.19 *	654.11 r

P_MSDAP/PISO_L/piso_reg_reg[3]/D (DFFNHX2M) 0.00 * 654.11 r
data arrival time 654.11

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.33 653.33
P_MSDAP/PISO_L/piso_reg_reg[3]/CKN (DFFNHX2M) 0.00 653.33 f
library hold time 0.24 653.57
data required time 653.57
---
data required time 653.57
data arrival time -654.11
---
slack (MET) 0.55

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[37]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[37]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point Incr Path
---
clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[37]/CKN (DFFNHX2M) 0.00 653.45 f
P_MSDAP/PISO_R/piso_reg_reg[37]/Q (DFFNHX2M) 0.60 654.05 r
P_MSDAP/PISO_R/U65/Y (AO22X2M) 0.19 * 654.24 r
P_MSDAP/PISO_R/piso_reg_reg[37]/D (DFFNHX2M) 0.00 * 654.24 r
data arrival time 654.24

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[37]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.24
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)

Endpoint: P_MSDAP/PISO_R/piso_reg_reg[3]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[3]/Q (DFFNHX2M)	0.59	654.04 r
P_MSDAP/PISO_R/U71/Y (AO22X2M)	0.20 *	654.24 r
P_MSDAP/PISO_R/piso_reg_reg[3]/D (DFFNHX2M)	0.00 *	654.24 r
data arrival time		654.24
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[3]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.67
data required time		653.67
data required time		653.67
data arrival time		-654.24
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[0]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[0]/Q (DFFNHX2M)	0.60	654.05 r
P_MSDAP/PISO_R/U33/Y (AO22X2M)	0.19 *	654.24 r
P_MSDAP/PISO_R/piso_reg_reg[0]/D (DFFNHX2M)	0.00 *	654.24 r
data arrival time		654.24
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45

P_MSDAP/PISO_R/piso_reg_reg[0]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.24
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[29]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[29]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[29]/CKN (DFFNHX2M)	0.00	653.32 f
P_MSDAP/PISO_L/piso_reg_reg[29]/Q (DFFNHX2M)	0.61	653.93 r
P_MSDAP/PISO_L/U57/Y (AO22X2M)	0.19 *	654.12 r
P_MSDAP/PISO_L/piso_reg_reg[29]/D (DFFNHX2M)	0.00 *	654.12 r
data arrival time		654.12

clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[29]/CKN (DFFNHX2M)	0.00	653.32 f
library hold time	0.23	653.55
data required time		653.55
data required time		653.55
data arrival time		-654.12
slack (MET)		0.57

Startpoint: P_MSDAP/main_ctrl/data_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/main_ctrl/data_count_reg[0]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

```
Point                                                   Incr       Path
--------------------------------------------------------------------------
clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.47     653.47
P_MSDAP/main_ctrl/data_count_reg[0]/CKN (DFFNHX2M)      0.00     653.47 f
P_MSDAP/main_ctrl/data_count_reg[0]/Q (DFFNHX2M)        0.55     654.03 f
P_MSDAP/main_ctrl/r100/A[0] (MSDAP_controller_DW01_inc_2)
                                                        0.00     654.03 f
P_MSDAP/main_ctrl/r100/U1/Y (INVX2M)                    0.06 *   654.09 r
P_MSDAP/main_ctrl/r100/SUM[0] (MSDAP_controller_DW01_inc_2)
                                                        0.00     654.09 r
P_MSDAP/main_ctrl/U120/Y (OAI2BB2X1M)                   0.21 *   654.30 r
P_MSDAP/main_ctrl/data_count_reg[0]/D (DFFNHX2M)        0.00 *   654.30 r
data arrival time                                               654.30

clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.47     653.47
P_MSDAP/main_ctrl/data_count_reg[0]/CKN (DFFNHX2M)      0.00     653.47 f
library hold time                                       0.25     653.73
data required time                                              653.73
--------------------------------------------------------------------------
data required time                                              653.73
data arrival time                                              -654.30
--------------------------------------------------------------------------
slack (MET)                                                     0.57


Startpoint: P_MSDAP/PISO_L/piso_reg_reg[13]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[13]
          (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                                   Incr       Path
--------------------------------------------------------------------------
clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.33     653.33
P_MSDAP/PISO_L/piso_reg_reg[13]/CKN (DFFNHX2M)          0.00     653.33 f
P_MSDAP/PISO_L/piso_reg_reg[13]/Q (DFFNHX2M)            0.62     653.95 r
P_MSDAP/PISO_L/U41/Y (AO22X2M)                          0.19 *   654.14 r
P_MSDAP/PISO_L/piso_reg_reg[13]/D (DFFNHX2M)            0.00 *   654.14 r
data arrival time                                               654.14

clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.33     653.33
```

P_MSDAP/PISO_L/piso_reg_reg[13]/CKN (DFFNHX2M) 0.00 653.33 f
library hold time 0.24 653.57
data required time 653.57
---
data required time 653.57
data arrival time -654.14
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[11]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[11]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[11]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[11]/Q (DFFNHX2M)	0.60	654.05 r
P_MSDAP/PISO_R/U39/Y (AO22X2M)	0.20 *	654.25 r
P_MSDAP/PISO_R/piso_reg_reg[11]/D (DFFNHX2M)	0.00 *	654.25 r
data arrival time		654.25

clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[11]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.67
data required time		653.67
data required time		653.67
data arrival time		-654.25
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

```
Point                                                   Incr       Path
-----------------------------------------------------------------------------
clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.33     653.33
P_MSDAP/PISO_L/piso_reg_reg[8]/CKN (DFFNHX2M)           0.00     653.33 f
P_MSDAP/PISO_L/piso_reg_reg[8]/Q (DFFNHX2M)             0.62     653.95 r
P_MSDAP/PISO_L/U36/Y (AO22X2M)                          0.19 *   654.14 r
P_MSDAP/PISO_L/piso_reg_reg[8]/D (DFFNHX2M)             0.00 *   654.14 r
data arrival time                                             654.14

clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.33     653.33
P_MSDAP/PISO_L/piso_reg_reg[8]/CKN (DFFNHX2M)           0.00     653.33 f
library hold time                                       0.24     653.57
data required time                                               653.57
-----------------------------------------------------------------------------
data required time                                               653.57
data arrival time                                               -654.14
-----------------------------------------------------------------------------
slack (MET)                                                      0.57


Startpoint: P_MSDAP/PISO_L/piso_reg_reg[37]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[37]
          (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                                   Incr       Path
-----------------------------------------------------------------------------
clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.33     653.33
P_MSDAP/PISO_L/piso_reg_reg[37]/CKN (DFFNHX2M)          0.00     653.33 f
P_MSDAP/PISO_L/piso_reg_reg[37]/Q (DFFNHX2M)            0.62     653.95 r
P_MSDAP/PISO_L/U65/Y (AO22X2M)                          0.20 *   654.14 r
P_MSDAP/PISO_L/piso_reg_reg[37]/D (DFFNHX2M)            0.00 *   654.14 r
data arrival time                                             654.14

clock Dclk (fall edge)                                651.00     651.00
clock network delay (propagated)                        2.33     653.33
P_MSDAP/PISO_L/piso_reg_reg[37]/CKN (DFFNHX2M)          0.00     653.33 f
library hold time                                       0.24     653.57
data required time                                               653.57
-----------------------------------------------------------------------------
data required time                                               653.57
```

```
data arrival time                                  -654.14
-------------------------------------------------------------------------
slack (MET)                                         0.57


Startpoint: P_MSDAP/PISO_L/piso_reg_reg[16]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[16]
         (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                       Incr       Path
-------------------------------------------------------------------------
clock Dclk (fall edge)                   651.00     651.00
clock network delay (propagated)           2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[16]/CKN (DFFNHX2M)    0.00     653.32 f
P_MSDAP/PISO_L/piso_reg_reg[16]/Q (DFFNHX2M)      0.61     653.93 r
P_MSDAP/PISO_L/U44/Y (AO22X2M)            0.20 *   654.12 r
P_MSDAP/PISO_L/piso_reg_reg[16]/D (DFFNHX2M)      0.00 *   654.12 r
data arrival time                                  654.12

clock Dclk (fall edge)                   651.00     651.00
clock network delay (propagated)           2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[16]/CKN (DFFNHX2M)    0.00     653.32 f
library hold time                        0.23     653.55
data required time                                 653.55
-------------------------------------------------------------------------
data required time                                 653.55
data arrival time                                 -654.12
-------------------------------------------------------------------------
slack (MET)                                        0.57


Startpoint: P_MSDAP/PISO_L/piso_reg_reg[22]
          (falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[22]
         (falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point                                       Incr       Path
-------------------------------------------------------------------------
clock Dclk (fall edge)                   651.00     651.00
clock network delay (propagated)           2.32     653.32
P_MSDAP/PISO_L/piso_reg_reg[22]/CKN (DFFNHX2M)    0.00     653.32 f
```

P_MSDAP/PISO_L/piso_reg_reg[22]/Q (DFFNHX2M) 0.61 653.93 r
P_MSDAP/PISO_L/U50/Y (AO22X2M) 0.19 * 654.13 r
P_MSDAP/PISO_L/piso_reg_reg[22]/D (DFFNHX2M) 0.00 * 654.13 r
data arrival time 654.13

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[22]/CKN (DFFNHX2M) 0.00 653.32 f
library hold time 0.23 653.55
data required time 653.55
---
data required time 653.55
data arrival time -654.13
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[14]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[14]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point Incr Path
---
clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[14]/CKN (DFFNHX2M) 0.00 653.45 f
P_MSDAP/PISO_R/piso_reg_reg[14]/Q (DFFNHX2M) 0.60 654.06 r
P_MSDAP/PISO_R/U42/Y (AO22X2M) 0.19 * 654.25 r
P_MSDAP/PISO_R/piso_reg_reg[14]/D (DFFNHX2M) 0.00 * 654.25 r
data arrival time 654.25

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.45 653.45
P_MSDAP/PISO_R/piso_reg_reg[14]/CKN (DFFNHX2M) 0.00 653.45 f
library hold time 0.22 653.67
data required time 653.67
---
data required time 653.67
data arrival time -654.25
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[8]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[8]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[8]/Q (DFFNHX2M)	0.60	654.06 r
P_MSDAP/PISO_R/U36/Y (AO22X2M)	0.19 *	654.25 r
P_MSDAP/PISO_R/piso_reg_reg[8]/D (DFFNHX2M)	0.00 *	654.25 r
data arrival time		654.25
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[8]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.68
data required time		653.68
data required time		653.68
data arrival time		-654.25
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[21]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[21]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[21]/CKN (DFFNHX2M)	0.00	653.32 f
P_MSDAP/PISO_L/piso_reg_reg[21]/Q (DFFNHX2M)	0.61	653.93 r
P_MSDAP/PISO_L/U49/Y (AO22X2M)	0.19 *	654.13 r
P_MSDAP/PISO_L/piso_reg_reg[21]/D (DFFNHX2M)	0.00 *	654.13 r
data arrival time		654.13

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[21]/CKN (DFFNHX2M) 0.00 653.32 f
library hold time 0.23 653.55
data required time 653.55
---
data required time 653.55
data arrival time -654.13
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[19]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[19]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[19]/CKN (DFFNHX2M)	0.00	653.32 f
P_MSDAP/PISO_L/piso_reg_reg[19]/Q (DFFNHX2M)	0.61	653.93 r
P_MSDAP/PISO_L/U47/Y (AO22X2M)	0.19 *	654.13 r
P_MSDAP/PISO_L/piso_reg_reg[19]/D (DFFNHX2M)	0.00 *	654.13 r
data arrival time		654.13

clock Dclk (fall edge) 651.00 651.00
clock network delay (propagated) 2.32 653.32
P_MSDAP/PISO_L/piso_reg_reg[19]/CKN (DFFNHX2M) 0.00 653.32 f
library hold time 0.23 653.55
data required time 653.55
---
data required time 653.55
data arrival time -654.13
---
slack (MET) 0.57

Startpoint: P_MSDAP/PISO_L/piso_reg_reg[25]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_L/piso_reg_reg[25]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk

Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[25]/CKN (DFFNHX2M)	0.00	653.32 f
P_MSDAP/PISO_L/piso_reg_reg[25]/Q (DFFNHX2M)	0.61	653.94 r
P_MSDAP/PISO_L/U53/Y (AO22X2M)	0.19 *	654.13 r
P_MSDAP/PISO_L/piso_reg_reg[25]/D (DFFNHX2M)	0.00 *	654.13 r
data arrival time		654.13
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.32	653.32
P_MSDAP/PISO_L/piso_reg_reg[25]/CKN (DFFNHX2M)	0.00	653.32 f
library hold time	0.23	653.55
data required time		653.55
data required time		653.55
data arrival time		-654.13
slack (MET)		0.57

Startpoint: P_MSDAP/PISO_R/piso_reg_reg[10]
(falling edge-triggered flip-flop clocked by Dclk)
Endpoint: P_MSDAP/PISO_R/piso_reg_reg[10]
(falling edge-triggered flip-flop clocked by Dclk)
Path Group: Dclk
Path Type: min

Point	Incr	Path
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[10]/CKN (DFFNHX2M)	0.00	653.45 f
P_MSDAP/PISO_R/piso_reg_reg[10]/Q (DFFNHX2M)	0.60	654.06 r
P_MSDAP/PISO_R/U38/Y (AO22X2M)	0.19 *	654.25 r
P_MSDAP/PISO_R/piso_reg_reg[10]/D (DFFNHX2M)	0.00 *	654.25 r
data arrival time		654.25
clock Dclk (fall edge)	651.00	651.00
clock network delay (propagated)	2.45	653.45
P_MSDAP/PISO_R/piso_reg_reg[10]/CKN (DFFNHX2M)	0.00	653.45 f
library hold time	0.22	653.67
data required time		653.67

```
---------------------------------------------------------------------------
data required time                                 653.67
data arrival time                                 -654.25
---------------------------------------------------------------------------
slack (MET)                                        0.58


Startpoint: P_MSDAP/SIPO_uut/bit_count_reg[0]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/bit_count_reg[0]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                       Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                     15.00      15.00
clock network delay (propagated)            1.33      16.33
P_MSDAP/SIPO_uut/bit_count_reg[0]/CKN (DFFNSRHX1M)      0.00      16.33 f
P_MSDAP/SIPO_uut/bit_count_reg[0]/Q (DFFNSRHX1M)        0.83      17.16 f
P_MSDAP/SIPO_uut/U105/Y (NAND2X2M)                      0.15 *    17.31 r
P_MSDAP/SIPO_uut/bit_count_reg[0]/D (DFFNSRHX1M)        0.00 *    17.31 r
data arrival time                                  17.31

clock Sclk (fall edge)                     15.00      15.00
clock network delay (propagated)            1.33      16.33
P_MSDAP/SIPO_uut/bit_count_reg[0]/CKN (DFFNSRHX1M)      0.00      16.33 f
library hold time                          0.17       16.50
data required time                                    16.50
---------------------------------------------------------------------------
data required time                                    16.50
data arrival time                                    -17.31
---------------------------------------------------------------------------
slack (MET)                                           0.82


Startpoint: P_MSDAP/SIPO_uut/data_L_reg[4]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[4]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                       Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                     15.00      15.00
```

clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_L_reg[4]/CKN (DFFNSRHX1M) 0.00 16.33 f
P_MSDAP/SIPO_uut/data_L_reg[4]/Q (DFFNSRHX1M) 0.80 17.13 r
P_MSDAP/SIPO_uut/U110/Y (OAI2BB2X1M) 0.22 * 17.35 r
P_MSDAP/SIPO_uut/data_L_reg[4]/D (DFFNSRHX1M) 0.00 * 17.35 r
data arrival time 17.35

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_L_reg[4]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.35
---
slack (MET) 0.85

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[0]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[0]/CKN (DFFNSRHX1M) 0.00 16.34 f
P_MSDAP/SIPO_uut/data_L_reg[0]/Q (DFFNSRHX1M) 0.79 17.13 r
P_MSDAP/SIPO_uut/U85/Y (OAI2BB2X1M) 0.23 * 17.36 r
P_MSDAP/SIPO_uut/data_L_reg[0]/D (DFFNSRHX1M) 0.00 * 17.36 r
data arrival time 17.36

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.34 16.34
P_MSDAP/SIPO_uut/data_L_reg[0]/CKN (DFFNSRHX1M) 0.00 16.34 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.36
---
slack (MET) 0.86

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[9]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[9]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[9]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[9]/Q (DFFNSRHX1M)	0.80	17.14 r
P_MSDAP/SIPO_uut/U94/Y (OAI2BB2X1M)	0.23 *	17.37 r
P_MSDAP/SIPO_uut/data_L_reg[9]/D (DFFNSRHX1M)	0.00 *	17.37 r
data arrival time		17.37
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[9]/CKN (DFFNSRHX1M)	0.00	16.34 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.37
slack (MET)		0.86

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[5]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[5]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[5]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_L_reg[5]/Q (DFFNSRHX1M)	0.80	17.14 r
P_MSDAP/SIPO_uut/U119/Y (OAI2BB2X1M)	0.23 *	17.37 r
P_MSDAP/SIPO_uut/data_L_reg[5]/D (DFFNSRHX1M)	0.00 *	17.37 r

```
data arrival time                                                  17.37

clock Sclk (fall edge)                                  15.00      15.00
clock network delay (propagated)                         1.33      16.33
P_MSDAP/SIPO_uut/data_L_reg[5]/CKN (DFFNSRHX1M)          0.00      16.33 f
library hold time                                       0.17       16.50
data required time                                                 16.50
---------------------------------------------------------------------------
data required time                                                 16.50
data arrival time                                                 -17.37
---------------------------------------------------------------------------
slack (MET)                                                        0.87


Startpoint: P_MSDAP/SIPO_uut/data_L_reg[1]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[1]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                                   Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                                  15.00      15.00
clock network delay (propagated)                         1.34      16.34
P_MSDAP/SIPO_uut/data_L_reg[1]/CKN (DFFNSRHX1M)          0.00      16.34 f
P_MSDAP/SIPO_uut/data_L_reg[1]/Q (DFFNSRHX1M)            0.80      17.13 r
P_MSDAP/SIPO_uut/U92/Y (OAI2BB2X1M)                      0.24 *    17.37 r
P_MSDAP/SIPO_uut/data_L_reg[1]/D (DFFNSRHX1M)            0.00 *    17.37 r
data arrival time                                                  17.37

clock Sclk (fall edge)                                  15.00      15.00
clock network delay (propagated)                         1.34      16.34
P_MSDAP/SIPO_uut/data_L_reg[1]/CKN (DFFNSRHX1M)          0.00      16.34 f
library hold time                                       0.17       16.50
data required time                                                 16.50
---------------------------------------------------------------------------
data required time                                                 16.50
data arrival time                                                 -17.37
---------------------------------------------------------------------------
slack (MET)                                                        0.87


Startpoint: P_MSDAP/SIPO_uut/data_L_reg[11]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[11]
```

(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[11]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[11]/Q (DFFNSRHX1M)	0.80	17.14 r
P_MSDAP/SIPO_uut/U120/Y (OAI2BB2X1M)	0.24 *	17.38 r
P_MSDAP/SIPO_uut/data_L_reg[11]/D (DFFNSRHX1M)	0.00 *	17.38 r
data arrival time		17.38
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[11]/CKN (DFFNSRHX1M)	0.00	16.34 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.38
slack (MET)		0.87

Startpoint: P_MSDAP/SIPO_uut/data_R_reg[9]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[9]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[9]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_R_reg[9]/Q (DFFNSRHX1M)	0.81	17.15 r
P_MSDAP/SIPO_uut/U91/Y (OAI2BB2X1M)	0.23 *	17.37 r
P_MSDAP/SIPO_uut/data_R_reg[9]/D (DFFNSRHX1M)	0.00 *	17.37 r
data arrival time		17.37
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[9]/CKN (DFFNSRHX1M)	0.00	16.33 f

library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.37
---
slack (MET) 0.88

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[13]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[13]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[13]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_L_reg[13]/Q (DFFNSRHX1M)	0.82	17.15 r
P_MSDAP/SIPO_uut/U121/Y (OAI2BB2X1M)	0.23 *	17.38 r
P_MSDAP/SIPO_uut/data_L_reg[13]/D (DFFNSRHX1M)	0.00 *	17.38 r
data arrival time		17.38
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_L_reg[13]/CKN (DFFNSRHX1M)	0.00	16.33 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.38
slack (MET)		0.88

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[7]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[7]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path

```
---------------------------------------------------------------------------
clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.34      16.34
P_MSDAP/SIPO_uut/data_L_reg[7]/CKN (DFFNSRHX1M)        0.00      16.34 f
P_MSDAP/SIPO_uut/data_L_reg[7]/Q (DFFNSRHX1M)          0.81      17.15 r
P_MSDAP/SIPO_uut/U93/Y (OAI2BB2X1M)              0.23 *    17.38 r
P_MSDAP/SIPO_uut/data_L_reg[7]/D (DFFNSRHX1M)          0.00 *    17.38 r
data arrival time                                       17.38

clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.34      16.34
P_MSDAP/SIPO_uut/data_L_reg[7]/CKN (DFFNSRHX1M)        0.00      16.34 f
library hold time                               0.17      16.50
data required time                                      16.50
---------------------------------------------------------------------------
data required time                                      16.50
data arrival time                                      -17.38
---------------------------------------------------------------------------
slack (MET)                                             0.88


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[0]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[0]
         (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                           Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[0]/CKN (DFFNSRHX1M)        0.00      16.33 f
P_MSDAP/SIPO_uut/data_R_reg[0]/Q (DFFNSRHX1M)          0.82      17.15 r
P_MSDAP/SIPO_uut/U81/Y (OAI2BB2X1M)              0.23 *    17.38 r
P_MSDAP/SIPO_uut/data_R_reg[0]/D (DFFNSRHX1M)          0.00 *    17.38 r
data arrival time                                       17.38

clock Sclk (fall edge)                          15.00      15.00
clock network delay (propagated)                 1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[0]/CKN (DFFNSRHX1M)        0.00      16.33 f
library hold time                               0.17      16.50
data required time                                      16.50
---------------------------------------------------------------------------
data required time                                      16.50
data arrival time                                      -17.38
```

```
---------------------------------------------------------------------------
slack (MET)                                      0.88


Startpoint: P_MSDAP/SIPO_uut/data_L_reg[3]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[3]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                    Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                  15.00      15.00
clock network delay (propagated)         1.33      16.33
P_MSDAP/SIPO_uut/data_L_reg[3]/CKN (DFFNSRHX1M)      0.00      16.33 f
P_MSDAP/SIPO_uut/data_L_reg[3]/Q (DFFNSRHX1M)        0.81      17.14 r
P_MSDAP/SIPO_uut/U118/Y (OAI2BB2X1M)                 0.24 *    17.38 r
P_MSDAP/SIPO_uut/data_L_reg[3]/D (DFFNSRHX1M)        0.00 *    17.38 r
data arrival time                                    17.38

clock Sclk (fall edge)                  15.00      15.00
clock network delay (propagated)         1.33      16.33
P_MSDAP/SIPO_uut/data_L_reg[3]/CKN (DFFNSRHX1M)      0.00      16.33 f
library hold time                        0.17      16.50
data required time                                 16.50
---------------------------------------------------------------------------
data required time                                 16.50
data arrival time                                 -17.38
---------------------------------------------------------------------------
slack (MET)                                        0.88


Startpoint: P_MSDAP/SIPO_uut/data_R_reg[5]
          (falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[5]
          (falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point                                    Incr       Path
---------------------------------------------------------------------------
clock Sclk (fall edge)                  15.00      15.00
clock network delay (propagated)         1.33      16.33
P_MSDAP/SIPO_uut/data_R_reg[5]/CKN (DFFNSRHX1M)      0.00      16.33 f
P_MSDAP/SIPO_uut/data_R_reg[5]/Q (DFFNSRHX1M)        0.81      17.14 r
```

P_MSDAP/SIPO_uut/U115/Y (OAI2BB2X1M) 0.24 * 17.39 r
P_MSDAP/SIPO_uut/data_R_reg[5]/D (DFFNSRHX1M) 0.00 * 17.39 r
data arrival time 17.39

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_R_reg[5]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.39
---
slack (MET) 0.89

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[2]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[2]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_L_reg[2]/CKN (DFFNSRHX1M) 0.00 16.33 f
P_MSDAP/SIPO_uut/data_L_reg[2]/Q (DFFNSRHX1M) 0.82 17.16 r
P_MSDAP/SIPO_uut/U86/Y (OAI2BB2X1M) 0.23 * 17.39 r
P_MSDAP/SIPO_uut/data_L_reg[2]/D (DFFNSRHX1M) 0.00 * 17.39 r
data arrival time 17.39

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_L_reg[2]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.39
---
slack (MET) 0.89

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[6]

(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[6]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[6]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[6]/Q (DFFNSRHX1M)	0.83	17.16 r
P_MSDAP/SIPO_uut/U111/Y (OAI2BB2X1M)	0.23 *	17.40 r
P_MSDAP/SIPO_uut/data_L_reg[6]/D (DFFNSRHX1M)	0.00 *	17.40 r
data arrival time		17.40

clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[6]/CKN (DFFNSRHX1M)	0.00	16.34 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.40
slack (MET)		0.89

Startpoint: P_MSDAP/SIPO_uut/bit_count_reg[3]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/bit_count_reg[3]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/bit_count_reg[3]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/bit_count_reg[3]/Q (DFFNSRHX1M)	0.77	17.10 r
P_MSDAP/SIPO_uut/U75/Y (XOR2X2M)	0.19 *	17.29 f
P_MSDAP/SIPO_uut/U51/Y (NAND2X2M)	0.10 *	17.39 r
P_MSDAP/SIPO_uut/bit_count_reg[3]/D (DFFNSRHX1M)	0.00 *	17.39 r
data arrival time		17.39

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/bit_count_reg[3]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.39
---
slack (MET) 0.90

Startpoint: P_MSDAP/SIPO_uut/data_R_reg[15]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[15]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point Incr Path
---
clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_R_reg[15]/CKN (DFFNSRHX1M) 0.00 16.33 f
P_MSDAP/SIPO_uut/data_R_reg[15]/Q (DFFNSRHX1M) 0.83 17.16 r
P_MSDAP/SIPO_uut/U95/Y (OAI2BB2X1M) 0.24 * 17.40 r
P_MSDAP/SIPO_uut/data_R_reg[15]/D (DFFNSRHX1M) 0.00 * 17.40 r
data arrival time 17.40

clock Sclk (fall edge) 15.00 15.00
clock network delay (propagated) 1.33 16.33
P_MSDAP/SIPO_uut/data_R_reg[15]/CKN (DFFNSRHX1M) 0.00 16.33 f
library hold time 0.17 16.50
data required time 16.50
---
data required time 16.50
data arrival time -17.40
---
slack (MET) 0.90

Startpoint: P_MSDAP/SIPO_uut/data_R_reg[13]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[13]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk

Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[13]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_R_reg[13]/Q (DFFNSRHX1M)	0.84	17.17 r
P_MSDAP/SIPO_uut/U117/Y (OAI2BB2X1M)	0.23 *	17.41 r
P_MSDAP/SIPO_uut/data_R_reg[13]/D (DFFNSRHX1M)	0.00 *	17.41 r
data arrival time		17.41
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[13]/CKN (DFFNSRHX1M)	0.00	16.33 f
library hold time	0.17	16.50
data required time		16.50
data required time		16.50
data arrival time		-17.41
slack (MET)		0.91

Startpoint: P_MSDAP/SIPO_uut/data_R_reg[1]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_R_reg[1]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[1]/CKN (DFFNSRHX1M)	0.00	16.33 f
P_MSDAP/SIPO_uut/data_R_reg[1]/Q (DFFNSRHX1M)	0.84	17.17 r
P_MSDAP/SIPO_uut/U89/Y (OAI2BB2X1M)	0.24 *	17.41 r
P_MSDAP/SIPO_uut/data_R_reg[1]/D (DFFNSRHX1M)	0.00 *	17.41 r
data arrival time		17.41
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.33	16.33
P_MSDAP/SIPO_uut/data_R_reg[1]/CKN (DFFNSRHX1M)	0.00	16.33 f
library hold time	0.17	16.50
data required time		16.50

---
data required time 16.50
data arrival time -17.41
---
slack (MET) 0.91

Startpoint: P_MSDAP/SIPO_uut/data_L_reg[12]
(falling edge-triggered flip-flop clocked by Sclk)
Endpoint: P_MSDAP/SIPO_uut/data_L_reg[12]
(falling edge-triggered flip-flop clocked by Sclk)
Path Group: Sclk
Path Type: min

Point	Incr	Path
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[12]/CKN (DFFNSRHX1M)	0.00	16.34 f
P_MSDAP/SIPO_uut/data_L_reg[12]/Q (DFFNSRHX1M)	0.84	17.18 r
P_MSDAP/SIPO_uut/U112/Y (OAI2BB2X1M)	0.24 *	17.41 r
P_MSDAP/SIPO_uut/data_L_reg[12]/D (DFFNSRHX1M)	0.00 *	17.41 r
data arrival time		17.41
clock Sclk (fall edge)	15.00	15.00
clock network delay (propagated)	1.34	16.34
P_MSDAP/SIPO_uut/data_L_reg[12]/CKN (DFFNSRHX1M)	0.00	16.34 f
library hold time	0.17	16.50
data required time		16.50
---		
data required time		16.50
data arrival time		-17.41
---		
slack (MET)		0.91