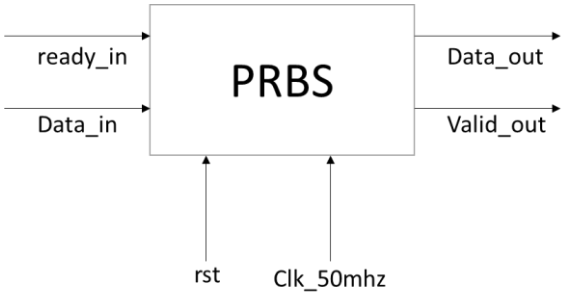
PRBS

Ports:

Signal	In /Out	Width	Description
Clk_50mhz	In	1	Input clock to the block of frequency 50MHz
rst	In	1	Reset
Ready_in	In	1	Signal to identify that input is ready
Data_in	In	1	Input data
Valid_out	Out	1	Signal to identify that output is valid
Data_out	Out	1	Output data

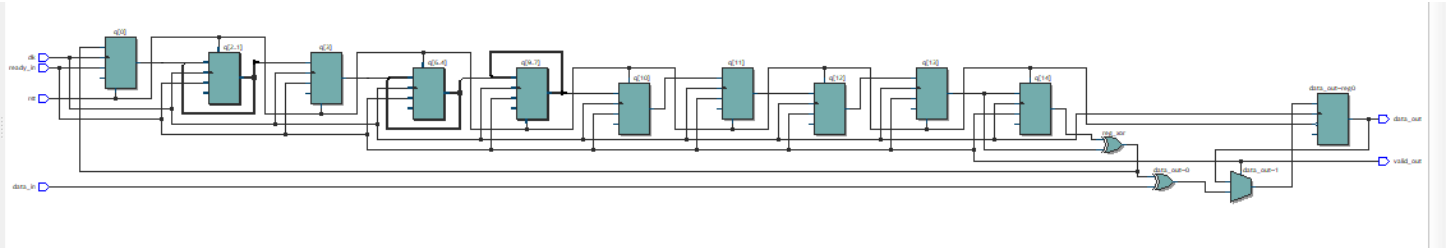
Block Diagram:



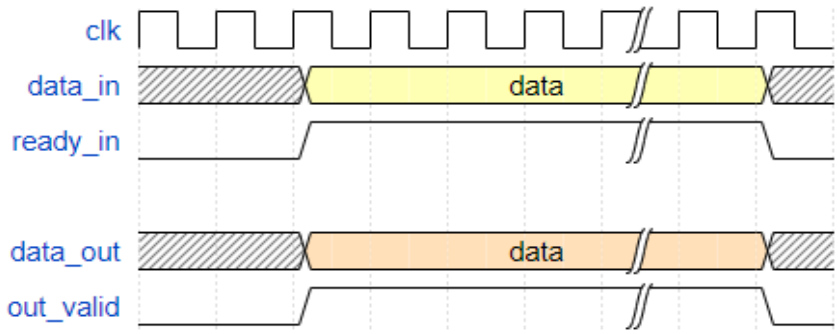
Timing and functionality:

- The handshaking signals are in phase with the input data
- No warmup cycles
- Total timing of the block = n cycles.

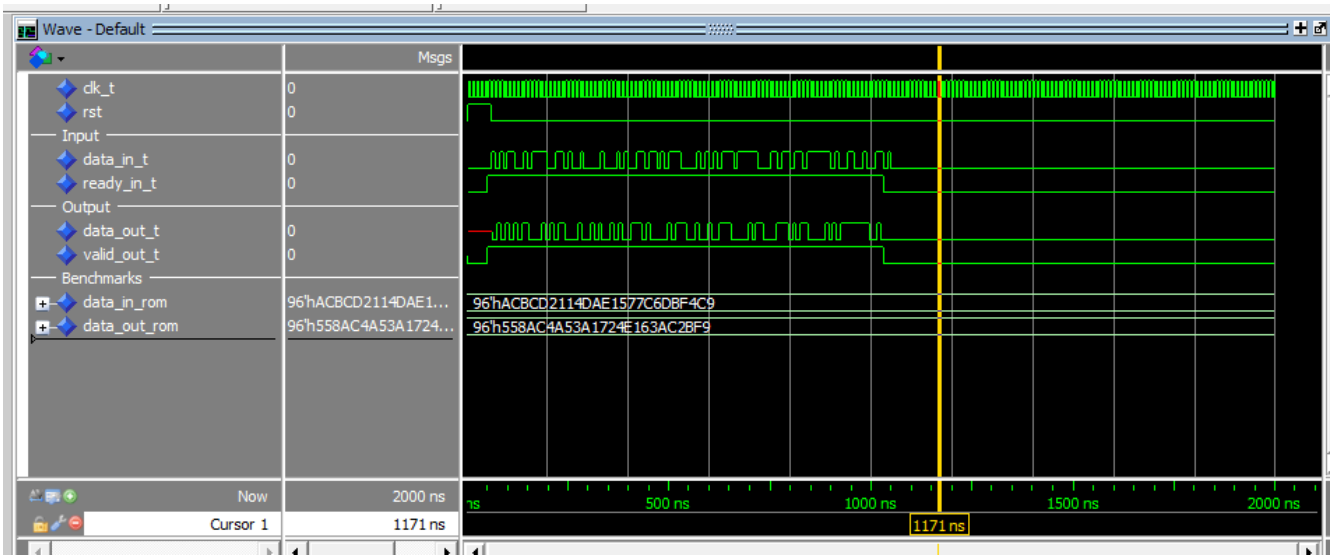
RTL:



Waveform:


clk
data_in
data
ready_in
data_out
data
out_valid

Results:

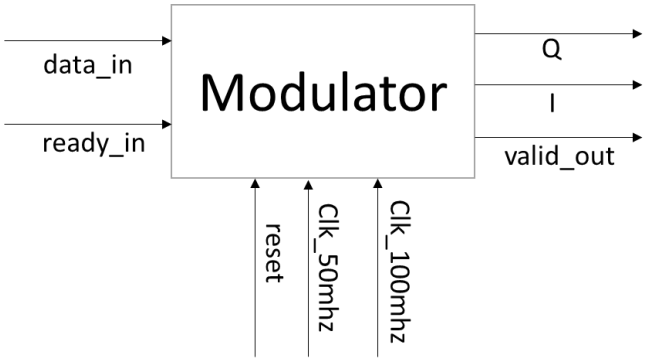

Wave - Default
Msgs
clk_t
rst
Input
data_in_t
ready_in_t
Output
data_out_t
valid_out_t
Benchmarks
data_in_rom
data_out_rom
0
0
0
0
0
0
96'hACBCD2114DAE1...
96'h558AC4A53A1724...
96'hACBCD2114DAE1577C6DBF4C9
96'h558AC4A53A1724E163AC2BF9
Now
2000 ns
Cursor 1
1171 ns
500 ns
1000 ns
1500 ns
2000 ns
1171 ns

Modulator

Ports:

Signal	In /Out	Width	Description
Clk_50mhz	In	1	Input clock to the block of frequency 50MHz
Clk_100mhz	In	1	Input clock to the block of frequency 100MHz
rst	In	1	Reset
ready_in	In	1	Signal to identify that input is ready
Data_in	In	1	Input data
Valid_out	Out	1	Signal to identify that output is valid
Q	Out	16	Output data Q
I	Out	16	Output data I

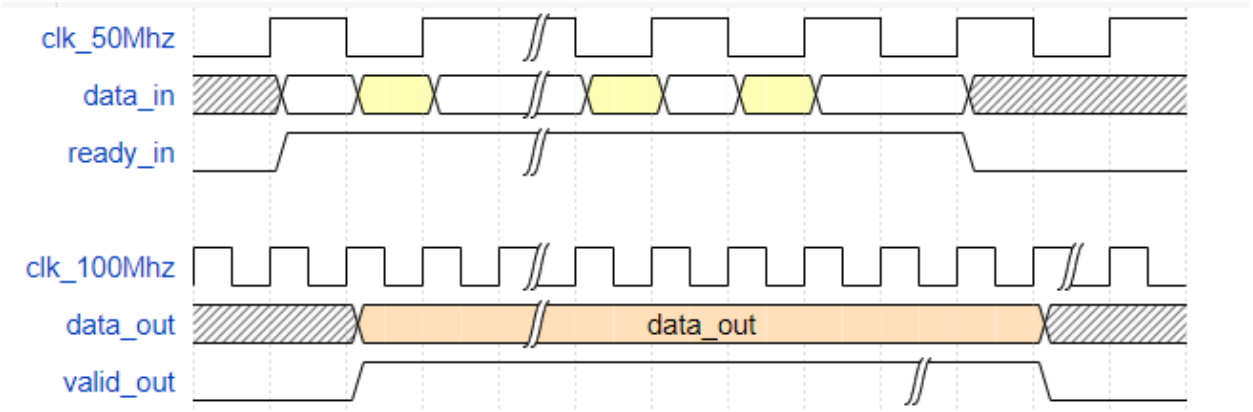
Block Diagram:



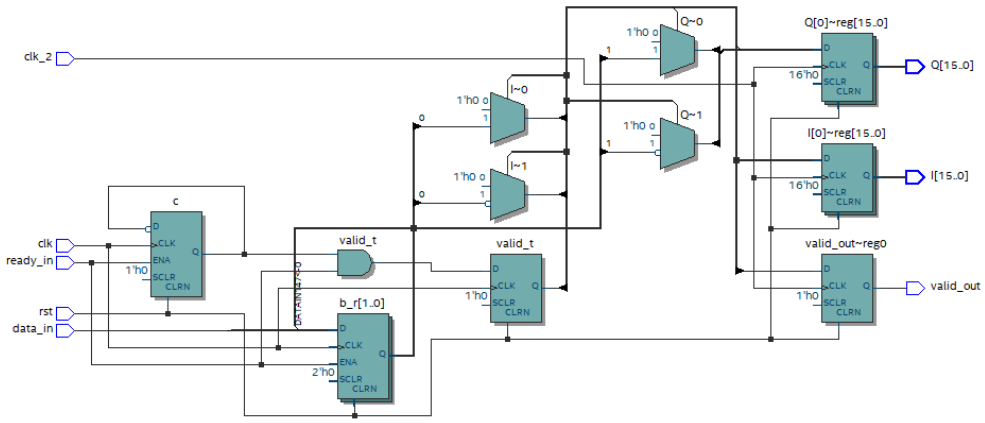
Timing and functionality:

- Output frequency is double the input frequency.
- Output bits and handshaking signals is late by two cycles (smaller clock frequency) due to the serialization of the input bits.
- Thus, total timing of the block = n + 2 cycles, where n is the number of bits.

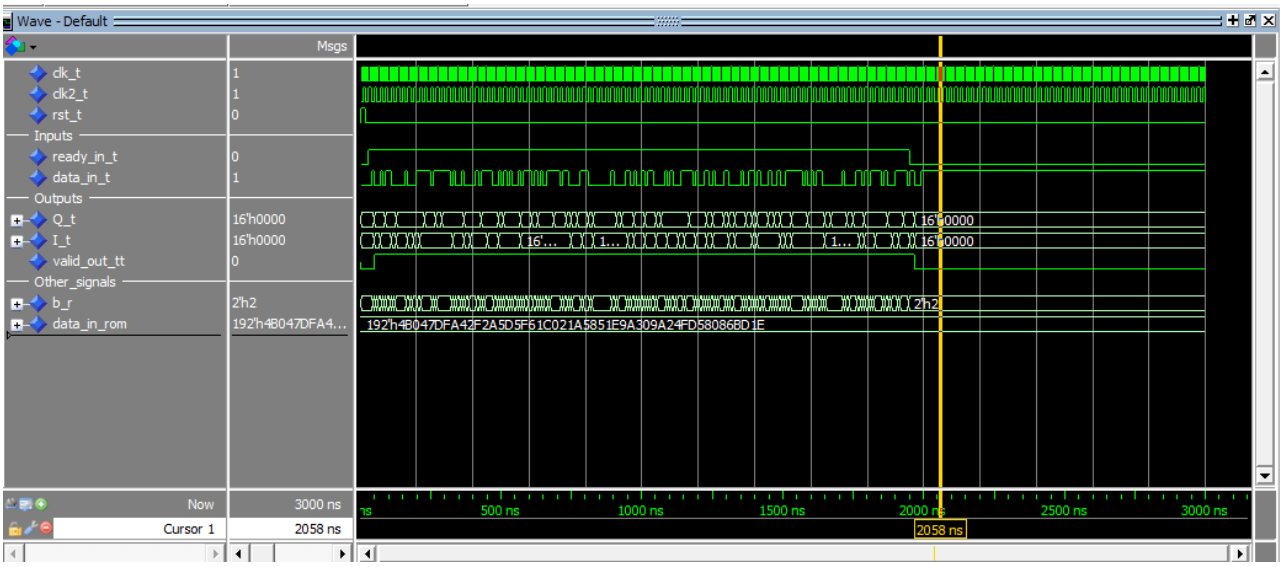
Wavefom:

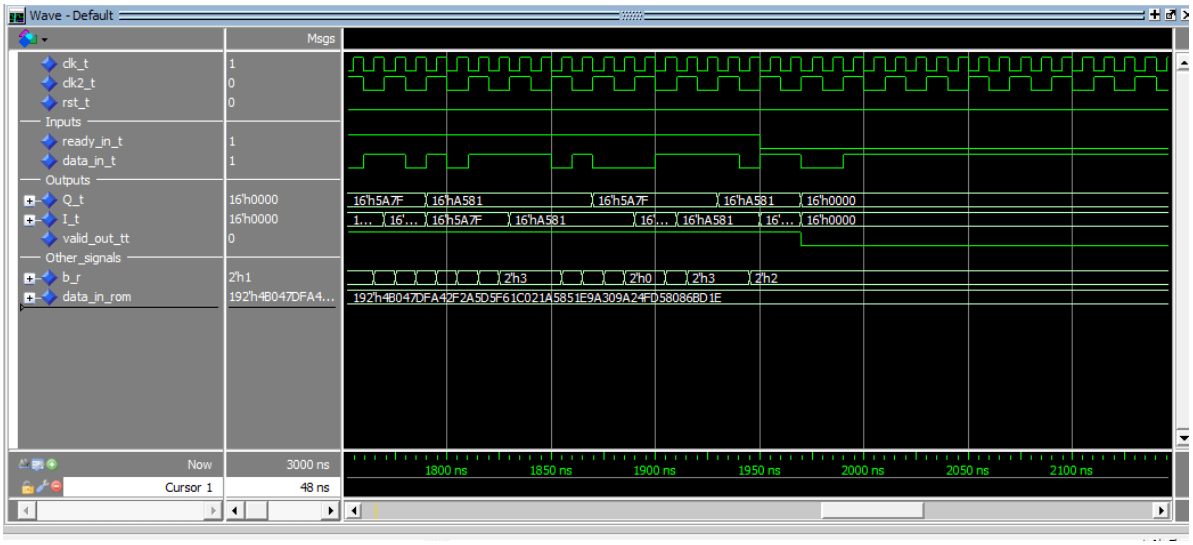

clk_50Mhz
data_in
ready_in
clk_100Mhz
data_out
data_out
valid_out

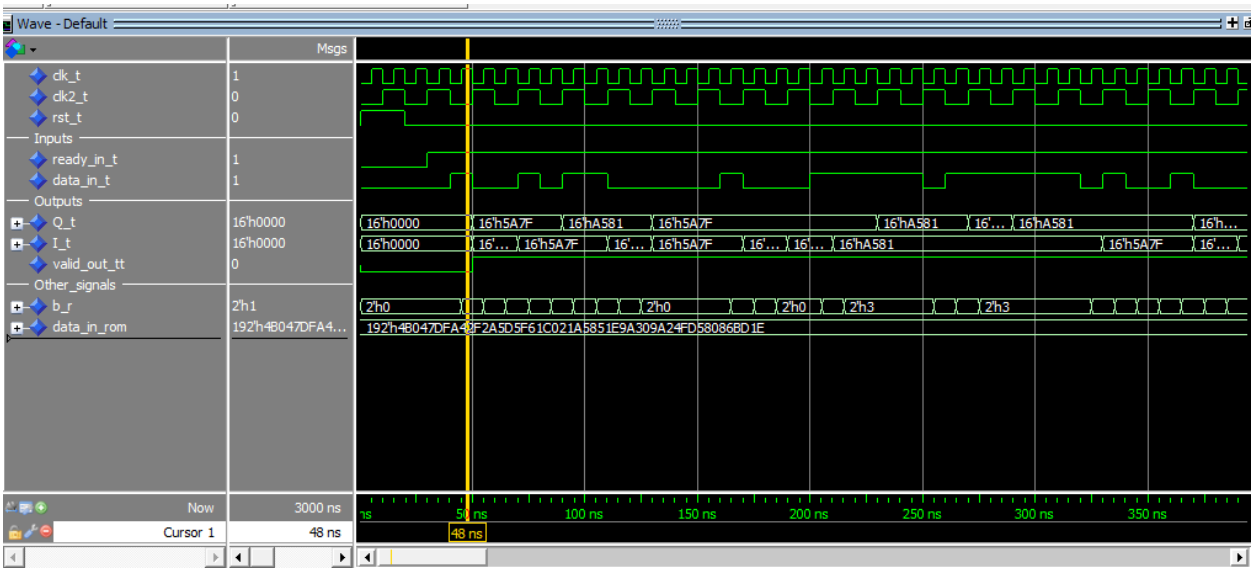
RTL


clk_2
clk
ready_in
rst
data_in
c
valid_t
valid_t
b_r[1..0]
I~0
I~1
Q~0
Q~1
Q[0]~reg[15..0]
I[0]~reg[15..0]
valid_out~reg0
Q[15..0]
I[15..0]
valid_out

Results:


Wave - Default
Msgs
clk_t
clk2_t
rst_t
Inputs
ready_in_t
data_in_t
Outputs
Q_t
I_t
valid_out_tt
Other_signals
b_r
data_in_rom
Now
Cursor 1
3000 ns
2058 ns
500 ns
1000 ns
1500 ns
2000 ns
2500 ns
3000 ns


Wave - Default
Msgs
clk_t
clk2_t
rst_t
Inputs
ready_in_t
data_in_t
Outputs
Q_t
I_t
valid_out_tt
Other_signals
b_r
data_in_rom
Now
Cursor 1
3000 ns
48 ns
1800 ns
1850 ns
1900 ns
1950 ns
2000 ns
2050 ns
2100 ns

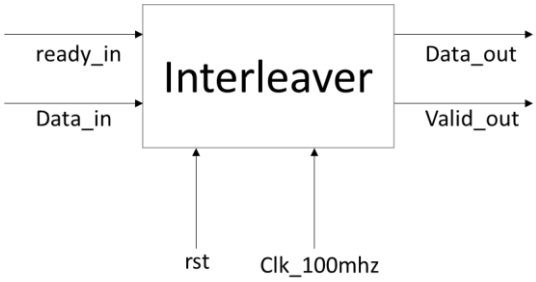

Wave - Default
Msgs
clk_t
clk2_t
rst_t
Inputs
ready_in_t
data_in_t
Outputs
Q_t
I_t
valid_out_tt
Other_signals
b_r
data_in_rom
Now
Cursor 1
3000 ns
48 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350 ns

Interleaver

Ports:

Signal	In /Out	Width	Description
Clk_100mhz	In	1	Input clock to the block of frequency 100MHz
rst	In	1	Reset
Ready_in	In	1	Signal to identify that input is ready
Data_in	In	1	Input data
Valid_out	Out	1	Signal to identify that output is valid
Data_out	Out	1	Output data

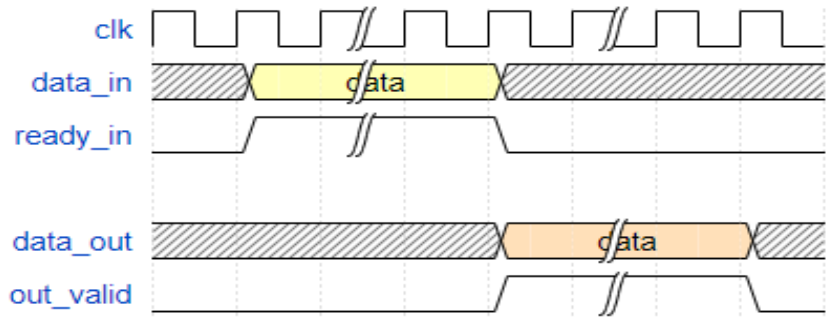
Block Diagram:



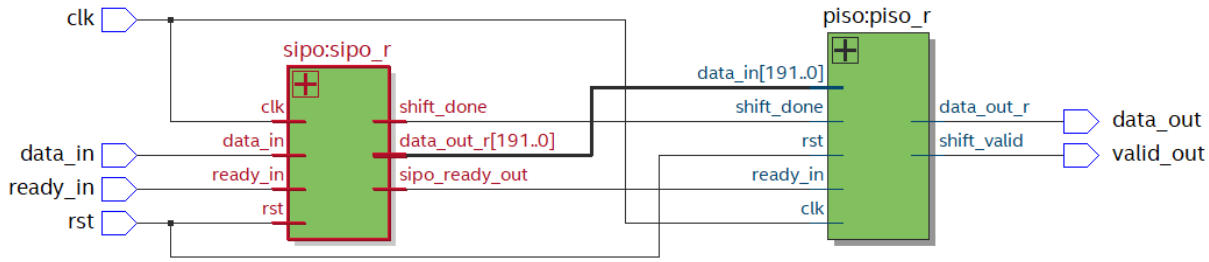
Timing and functionality

- The data are made parallel, then the wires were permuted and serialized
- Delay of serialization = n, where n is the number of bits
- Delay of interleaving = 0, it just wires re-ordering
- Delay of serialization = 2, one cycle for loading and one cycle for shifting. Since, the input is parallel. Thus, there is no output warmup cycles wasted
- Total delay of the block = n+2

Waveform:

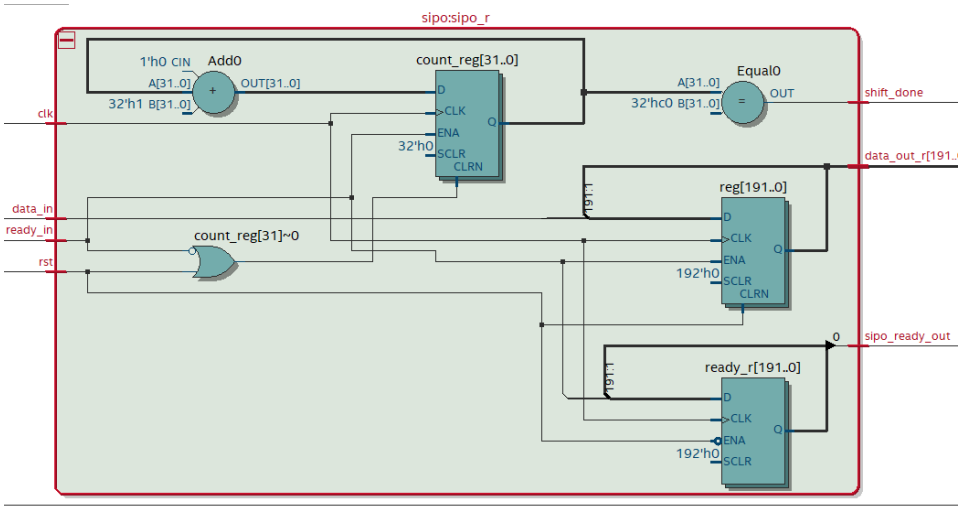


RTL:



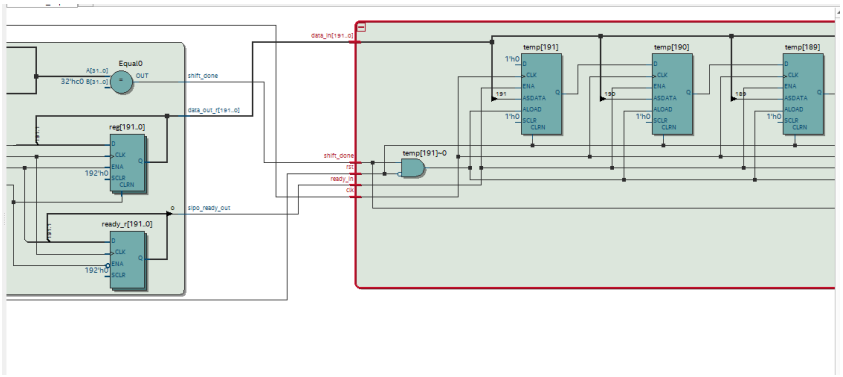
1- Parallelization

- Serial in Parallel output shift register



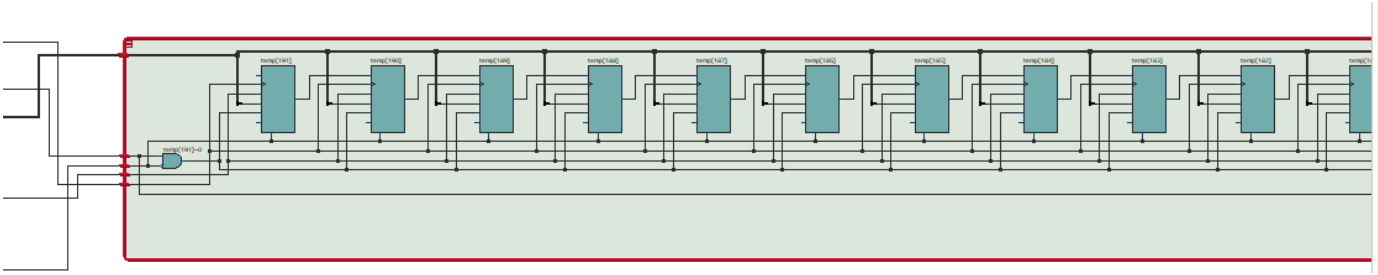
2- Permuting

- Simple rewiring between registers

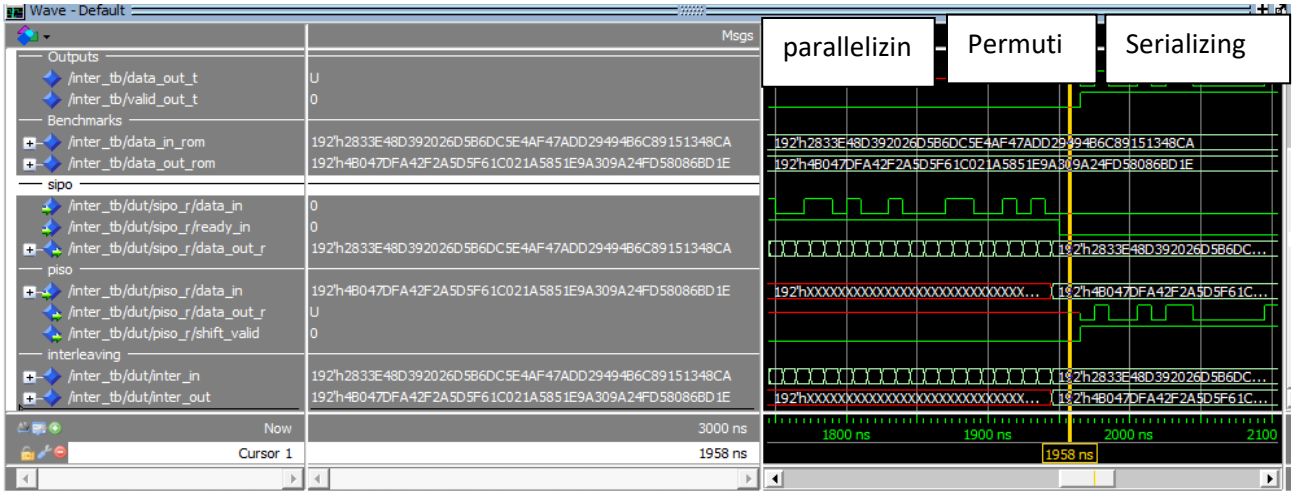


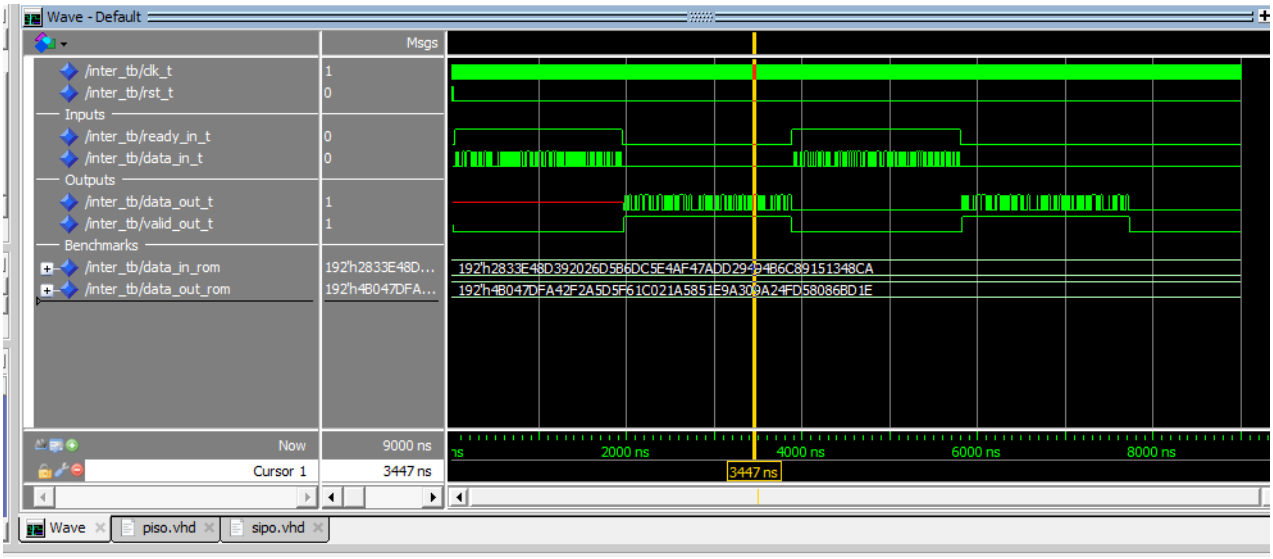
3- Serializing

- Parallel in Serial shift register



Results:



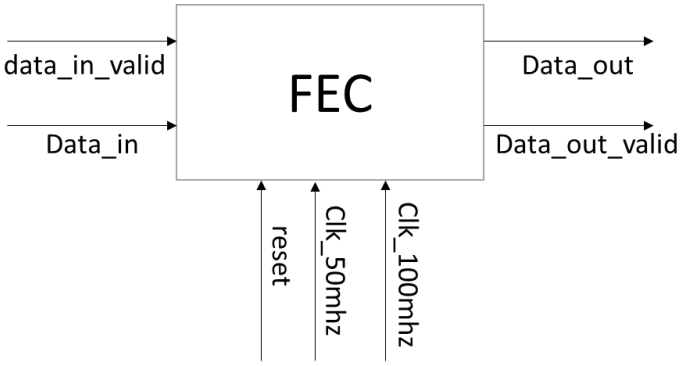

Wave - Default
Msgs
/inter_tb/clk_t
/inter_tb/rst_t
Inputs
/inter_tb/ready_in_t
/inter_tb/data_in_t
Outputs
/inter_tb/data_out_t
/inter_tb/valid_out_t
Benchmarks
/inter_tb/data_in_rom
/inter_tb/data_out_rom
1
0
0
0
1
1
192'h2833E48D...
192'h4B047DFA...
192'h2833E48D392026D5B6DC5E4AF47ADD2949B6C89151348CA
192'h4B047DFA42F2A5D5F61C021A5851E9A309A24FD58086BD1E
Now
9000 ns
Cursor 1
3447 ns
2000 ns
4000 ns
6000 ns
8000 ns
3447 ns
Wave
piso.vhd
sipo.vhd

FEC

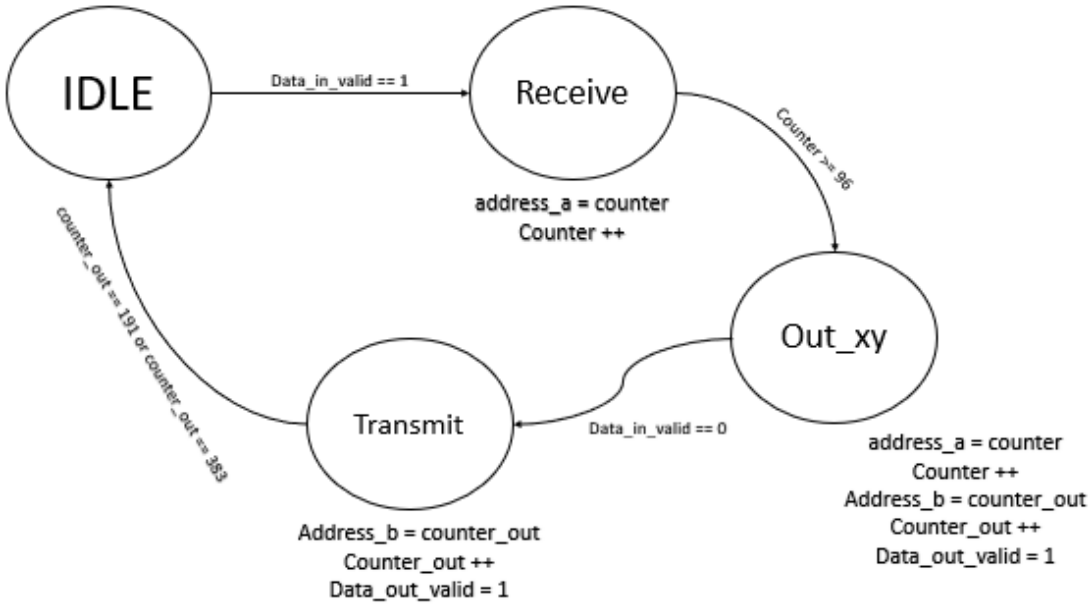
Ports:

Signal	In /Out	Width	Description
Clk_50mhz	In	1	Input clock to the block of frequency 50MHz
Clk_100mhz	In	1	Input clock to the block of frequency 100MHz
reset	In	1	Reset
Data_in_valid	In	1	Signal to identify that input is ready
Data_in	In	1	Input data
Data_out_valid	Out	1	Signal to identify that output is valid
Data_out	Out	1	Output data

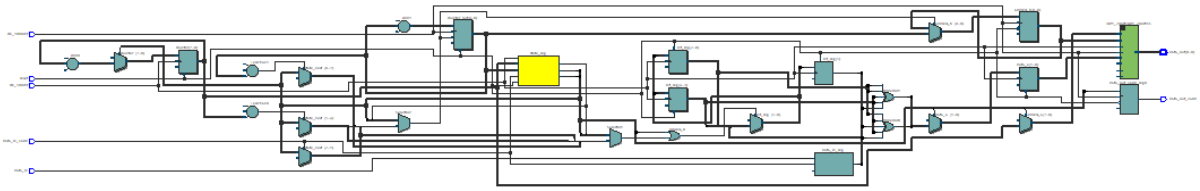
Block Diagram:



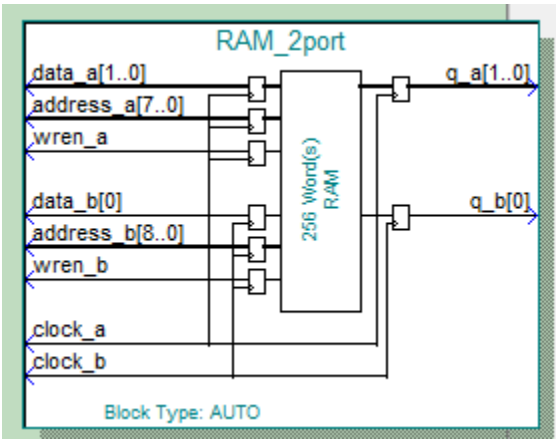
State Machine Diagram:



RTL:



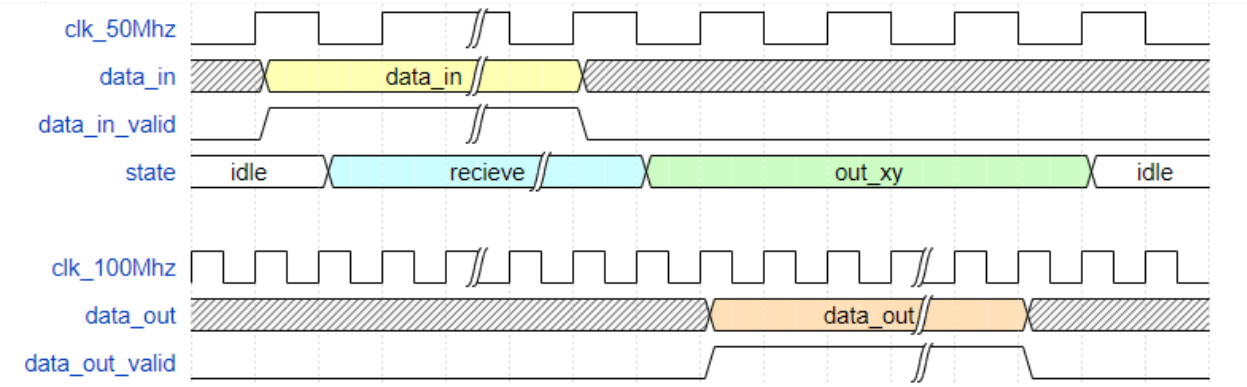
Memory interface:



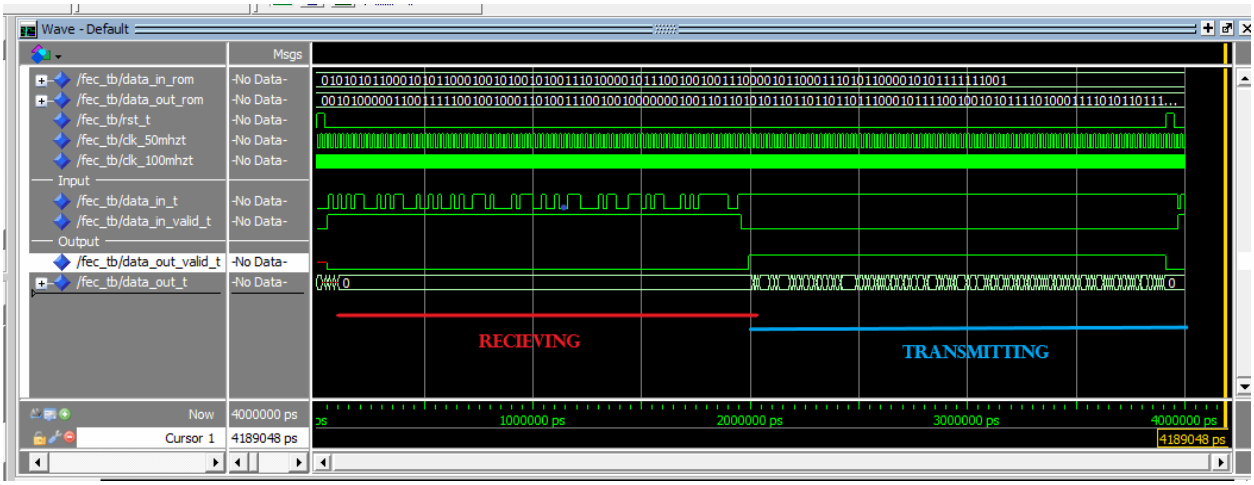
Timing and Functionality:

1- The data are received and processed in parallel then saved in dual-port RAM.
2- The output data is extracted from RAM after receiving and processing all the input bits.

Waveform:



Results:

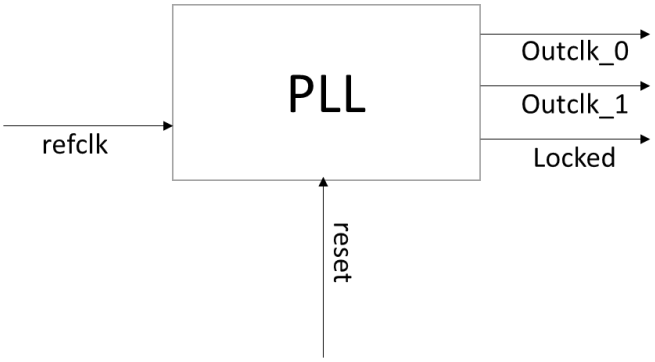


PLL

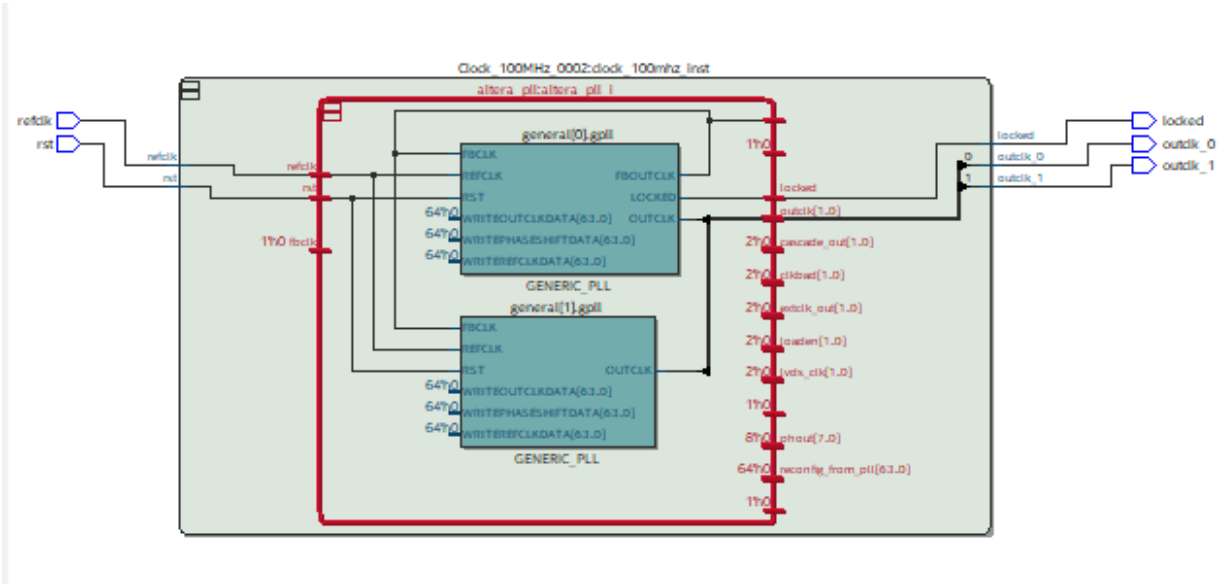
Ports:

Signal	In /Out	Description
refclk	In	Reference clock of frequency 50MHz
reset	In	Reset
locked	Out	Identifies when the signal is valid
Outclk_0	Out	Signal to identify that output is valid
Outclk_1	Out	Output data

Block Diagram:



RTL:



Phase 2

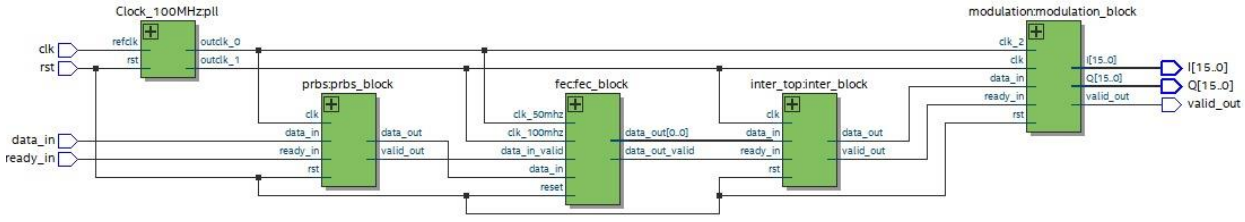
Top Module

Ports:

Signal	In /Out	Width	Description
Clk_50mhz	In	1	Input clock to the block of frequency 50MHz
Data_in	In	1	Input data
Ready_in	In	1	Signal to identify that input is ready
Rst	In	1	Reset
Valid_out	In	1	Signal to identify that output is valid
Q	Out	1	Output data
I	Out	1	Output data

Block Diagram:

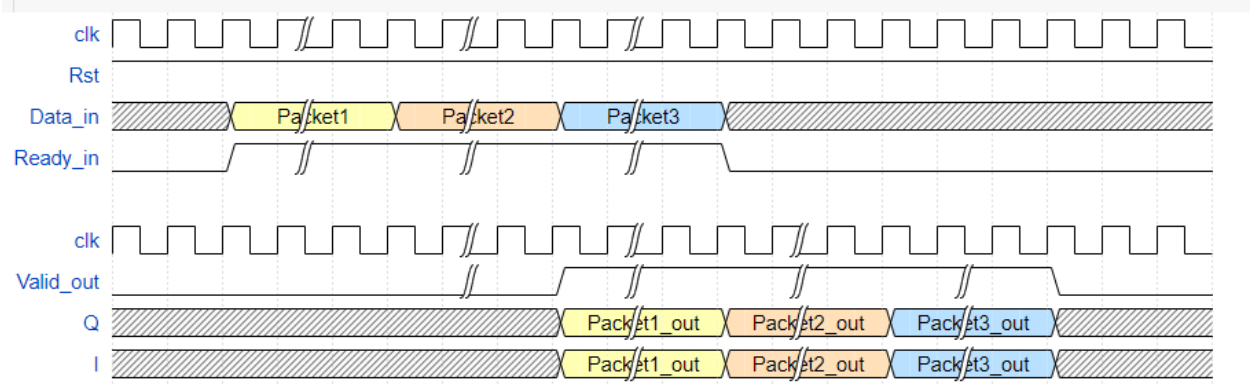
RTL:



Timing and Functionality:

1- The data are processed in the four blocks
2- The output stream is valid and continuous as long as the input stream is ready and continuous.

Waveform:



Results: