Politecnico di Torino

Master degree in Electrical Engineering

Laboratory session n°2
Design and implementation of a digital arithmetic

Integrated system architecture

Authors: Tesser Andrea, Vianello Nicola, Codazzo Stefano

December 17, 2019

Abstract

The goal of this laboratory session is to deal with the synthesis of behavioral digital arithmetic units and to design a 32 bit multiplier using a modified booth encoding and a Dadda tree.

In order to focus mainly on these aspects, the synthesis tests are performed on an already designed open-source floating point multiplier[*], which will be then modified substituting the behavioral unsigned multiplier with the one designed by us.

These tests consist in finding the maximum frequency and the area of the flattened design, first without forcing a specific implementation of the multiplier, and then forcing the compiler using a carry-save multiplier and a parallel-prefix multiplier. At this point, a register is added at the output of the multiplier in order to improve the pipeline and the *optimize-registers* command is issued first after a normal compilation, and then exploiting the Synopsys Design Compiler *ultra* mode.

[*]fpuvhdl from opencores.org

Contents

List of Figures

List of Tables

List of Listings

CHAPTER 1

Digital arithmetic and logic synthesis

The first step consists in designing a suitable test to verify the correctness of the floating point multiplier, whose diagram is show in figure 1.1.

In order to do this the Verilog testbench of listing B.1 is used where the following four VHDL modules are recalled, connected as in figure 1.2:

- Data Maker: this component has the task to read the samples from a file and to provide one of these per clock cycle to both the inputs of the device under test. When there are no more data to send, it rise a flag called *END_SIM* which will be read from the Data Sink module;
- DUT: the device under test;
- Data Sink: this is needed to write all the results generated by the DUT in a file. When the *END_SIM* signal is raised by the Data Maker it stops the writing of the samples, in this way meaningless results are discharged;
- Clock Generator: it is responsible to generate the clock signal for all the above modules.

Once the testbanch is ready, a simulation is performed using Mentor ModelSim giving the input file of listing B.5, the floating point multiplier compute their square value and the generated file is compared with the expected results, shown in listing B.6. In figure 1.3 it is possible to see the computation of these values. Since the pipeline has a latency of four clock cycles, the first four outputs are unknown.

Once the design is correctly tested, a register is added for both the input operands as in listing A.1, in order to have all the inputs sequential and to avoid problem with the synthesis tool. Then the simulation is performed again with the same procedure described before, in order to verify that all still work correctly. As can be seen in figure 1.4, the only difference between the previous simulation is about the latency that, as expected, is increased by one clock cycles due to the presence of the additional registers on the inputs.

At this point the design is ready to proceeds with the synthesis test. The significands multiplier present on the second stage is simply described by means of the ”*” operator, in this way Synopsys Design Compiler is free to implement the structure as it sees fit according to the given constraints, using the possible implementation available in Design Ware library. It is possible to force Design Compiler using a specific architecture using the command *set_implementation*. On this basis, three synthesis are performed to test different structures of the multiplier:

- First synthesis: no constraint on the type of multiplier (script of listing C.1);
- Second synthesis: CSA multiplier (script of listing C.2);

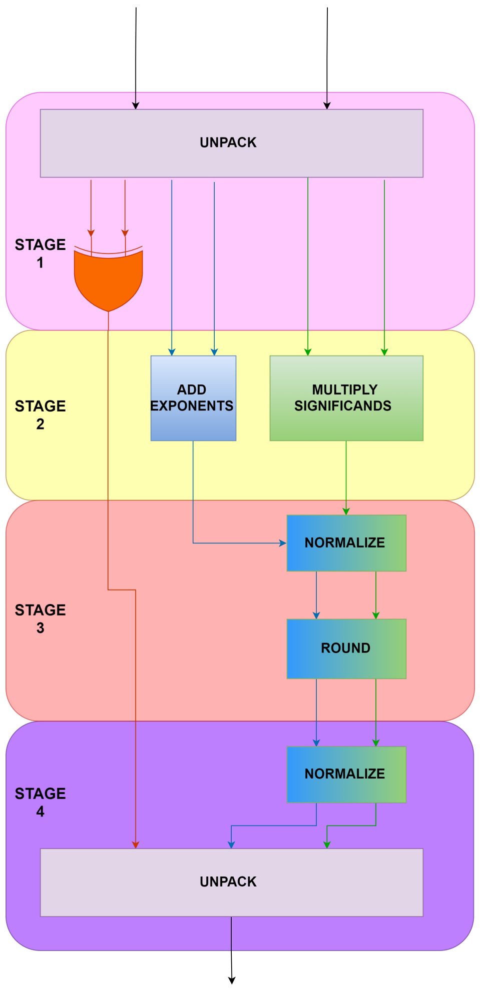


Figure 1.1: Floating point multiplier diagram.

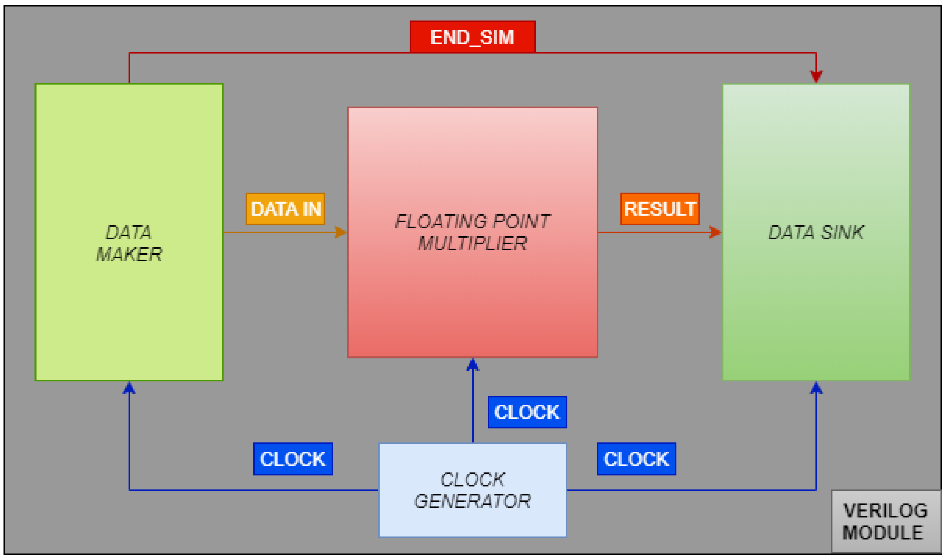


Figure 1.2: Testbench block diagram.

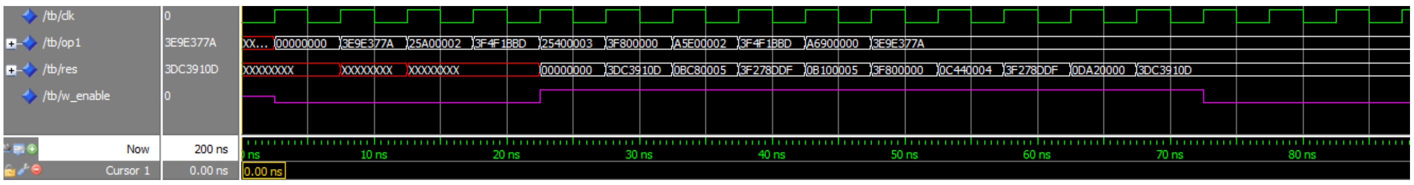

/tb/clk
/tb/op1
/tb/res
/tb/w_enable
0
3E9E377A
3DC3910D
0
XX... 00000000 3E9E377A 25A00002 3F4F1BBD 25400003 3F800000 A5E00002 3F4F1BBD A6900000 3E9E377A
XXXXXXXX XXXXXXXX XXXXXXXX 00000000 3DC3910D 0BC80005 3F278DDF 0B100005 3F800000 0C440004 3F278DDF 0DA20000 3DC3910D
Now 200 ns
Cursor 1 0.00 ns
0.00 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns

Figure 1.3: Simulation of the given design.

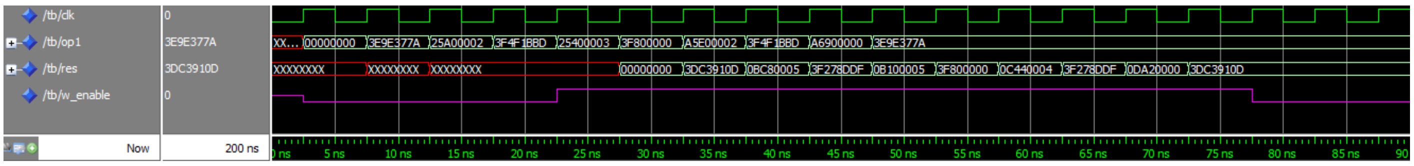

/tb/clk
/tb/op1
/tb/res
/tb/w_enable
0
3E9E377A
3DC3910D
0
XX... 00000000 3E9E377A 25A00002 3F4F1BBD 25400003 3F800000 A5E00002 3F4F1BBD A6900000 3E9E377A
XXXXXXXX XXXXXXXX XXXXXXXX 00000000 3DC3910D 0BC80005 3F278DDF 0B100005 3F800000 0C440004 3F278DDF 0DA20000 3DC3910D
Now 200 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
60 ns
65 ns
70 ns
75 ns
80 ns
85 ns
90

Figure 1.4: Simulation after adding a register for each input.

- Third synthesis: PPARCH multiplier (script of listing C.3).

The results obtained from the three synthesis are summarized in tables 1.1 and 1.2. For all the structures the maximum frequency and the corresponding area are found setting first a timing constraint of 0 ns, then repeating the synthesis with the slack found in the previous step (with positive sign), at this point if the constraint is met the maximum frequency is found, otherwise another synthesis is performed adding the new slack (with positive sign) to the previous timing constraint, and so on until the constraint turns out to be met. From these data it is possible to see that the CSA multiplier has the worst result in terms of both area and maximum frequency. Furthermore, when Design Compiler has no constraints on the type of structure, it uses the PPARCH implementation. In listings D.1, D.4, D.7 are reported for every synthesis the implementation of all the arithmetic blocks present in the design.

Implementation	**Comb area** [µm^2]	**BufInv area** [µm^2]	**Non-comb area** [µm^2]	**Total cell area** [µm^2]
Any	2861.9	232.2	1137.2	3999
CSA	3710.9	153.2	1140.4	4851.3
PPARCH	2849.9	223.7	1137.2	3987

Table 1.1: Area reports forcing a CSA multiplier, a PPARCH multiplier, and without constraints.

Implementation	**Clock Period** [ns]	**Max Frequency** [MHz]
Any	1.50	667
CSA	4.13	242
PPARCH	1.50	667

Table 1.2: Timing reports forcing a CSA multiplier, a PPARCH multiplier, and without constraints.

CHAPTER 2

Fine-grain Pipelining and optimization

The next test is about retiming: the internal structure of the design is modified adding one register at the output of the significand multiplier letting Design Compiler move it in the best position. Clearly, all the needed registers to ensure the correct timing of the stage 2 are added in the other paths, the final VHDL is reported in listing A.2.

Again, the ModelSim simulation is performed: as it can be seen in figure 2.1 all still work properly and as expected the latency is increased of one clock cycle.

Once the design is tested, some synthesis are performed in order to find the maximum frequency and the corresponding area as explained in the previous chapter, but now issuing the *optimize_registers* command after the synthesis. The used script is the one of listing C.4, which also save the timing and area reports both before and after the retiming. In table 2.1 it is possible to see a summary of these reports, the maximum frequency after the registers optimization becomes more than twice that we had before, at the cost of a bigger area.

At this point the retiming is tested again but replacing the normal *compile* command with *compile_ultra*. This enable some extra feature of Design Compiler, which increasing the computational time can reach a better results in therm of area and maximum frequency. The synthesis script is in listing C.5, while in table 2.2 is reported a comparison between the two commands: the maximum frequency becomes slightly higher and the area decreases.

The final step consists to design a 32 bit unsigned multiplier using a Modified Booth Encoding and a Dadda tree, and to use it as significand multiplier in the second stage, in place of the previous behavioral operator. The designing is described in the following chapter, so here only the synthesis results and a comparison with the previous implementations are reported. With this component two synthesis are performed: one without the added register at the output of the multiplier, and one with this register and retiming. In this way it is possible to make a comparison between both the result of these two chapter. The used scripts are respectively in listing C.6 and C.7. In the first case the obtained maximum frequency is equal to 667 MHz and the area is equal to 5115 μm^2, so it is always worse than the Design Wares implementations. In the second case the reports are: 1.69 GHz

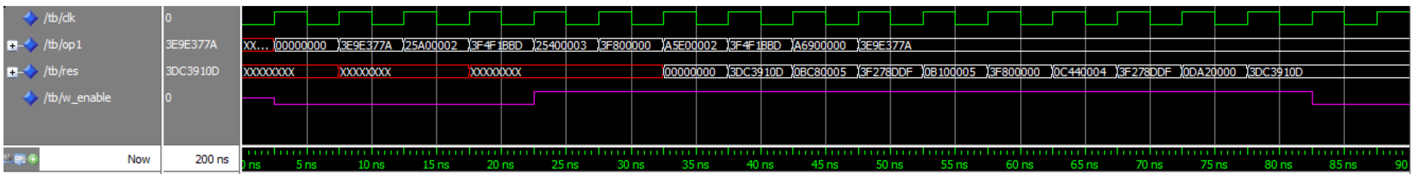

/tb/clk
/tb/op1
/tb/res
/tb/w_enable
Now
0
3E9E377A
3DC3910D
0
200 ns
XX... 00000000 3E9E377A 25A00002 3F4F1BBD 25400003 3F800000 A5E00002 3F4F1BBD A6900000 3E9E377A
XXXXXXXX XXXXXXXX XXXXXXXX 00000000 3DC3910D 0BC80005 3F278DDF 0B100005 3F800000 0C440004 3F278DDF 0DA20000 3DC3910D
0 ns 5 ns 10 ns 15 ns 20 ns 25 ns 30 ns 35 ns 40 ns 45 ns 50 ns 55 ns 60 ns 65 ns 70 ns 75 ns 80 ns 85 ns 90

Figure 2.1: Simulation after adding a register at the significand multiplier output.

and 7371 µm^2, therefore, even with the additional register, the provided arithmetic units library is favorable.

Registers	**Max frequency**	**Area** [µm^2]
Not optimized	769 MHz	4300
Optimized	1.67 GHz	5406

Table 2.1: Maximum frequency and area with and without optimizing the registers.

Command	**Max frequency** [GHz]	**Area** [µm^2]
compile	1.67	5406
compile_ultra	1.69	5246

Table 2.2: Maximum frequency and area with and without using *compil_ultra* command.

CHAPTER 3

MBE multiplier

As already mentioned, the designed 32 bit unsigned multiplier is based on a Dadda tree and a Modified Booth Encoding. This encoding uses a radix-4 approach, this means that calling A the multiplicand and B the multiplier, each partial product is generated taking three bits of B, so that the bits taken to compute the next partial product start from the last bit used to compute the previous partial product. In order to allow this type of encoding B is extended on 35 bits, adding a '0' in the positions -1, 33, 34. In this way seventeen partial products are generated according to the table 3.1, each of them has 33 bits because the multiplication by two corresponds to a left shift of one position.

Since the partial products can be negative and to compute the two's complement is necessary to negate all the bits and to add 1, in the partial products generation phase only the negation is performed and the information about the sign is propagated to the Dadda tree, which will add or not some 1s in the correct positions (this depend on the weight of the partial product) exploiting the tree itself. In figure 3.1a the partial products, their sign, their weight, and the bits of B used to compute them are represented.

In the Dadda tree some half-adders and full-adders are used to reduce the sum of the all partial products in a sum of only two numbers, using an "as late as possible" approach. In order to exploit this type of policy without increasing the depth of the tree, it is divided in stages and every stage has a maximum amount of lines that it is able to carry. Since every stage can compress three lines into two, starting from the stage 0 that is the one with two lines, we can compute that the stage 6 must have nineteen lines at most (see table 3.2). This depth is sufficient for our tree, that as already mentioned, has to sum seventeen partial products and so seventeen lines. In figure 3.1b, how the partial products compose the stage 6 is shown. Note that the bits of this stage and their relative weights are the same of figure 3.1a, but they are moved as upward as possible in order to fill as much as possible the

$\mathbf{B_{2i+1}B_{2i}B_{2i\text{-}1}}$	**Partial product i**
000	0
001	+A
010	+A
011	+2A
100	-2A
101	-A
110	-A
111	0

Table 3.1: Modified Booth Encoding.

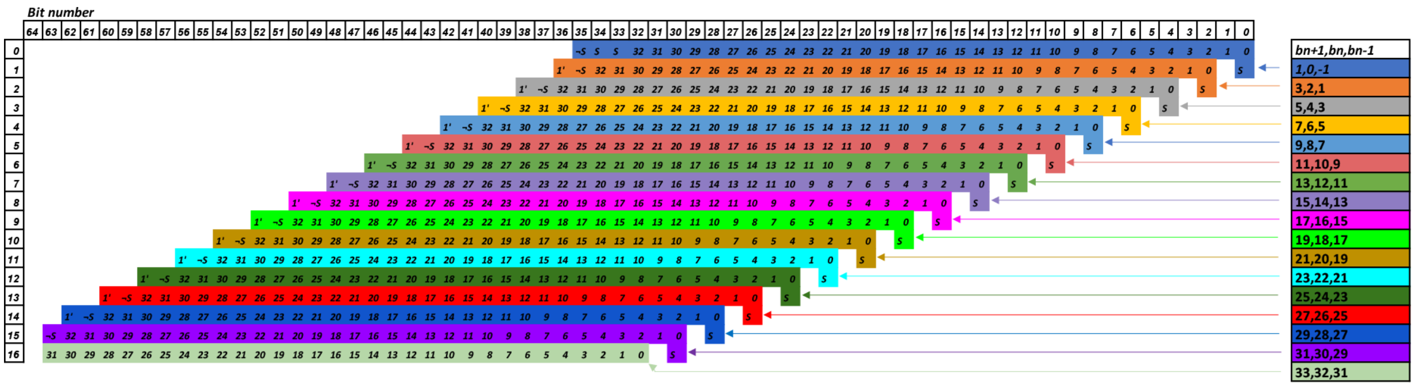
Stage	**Lines**	**FAs**	**HAs**
6	19	40	21
5	13	104	12
4	9	120	9
3	6	100	6
2	4	56	3
1	3	60	2
0	2	63	1
Total:		543	54

Table 3.2: Maximum number of lines for each stage and used resources.

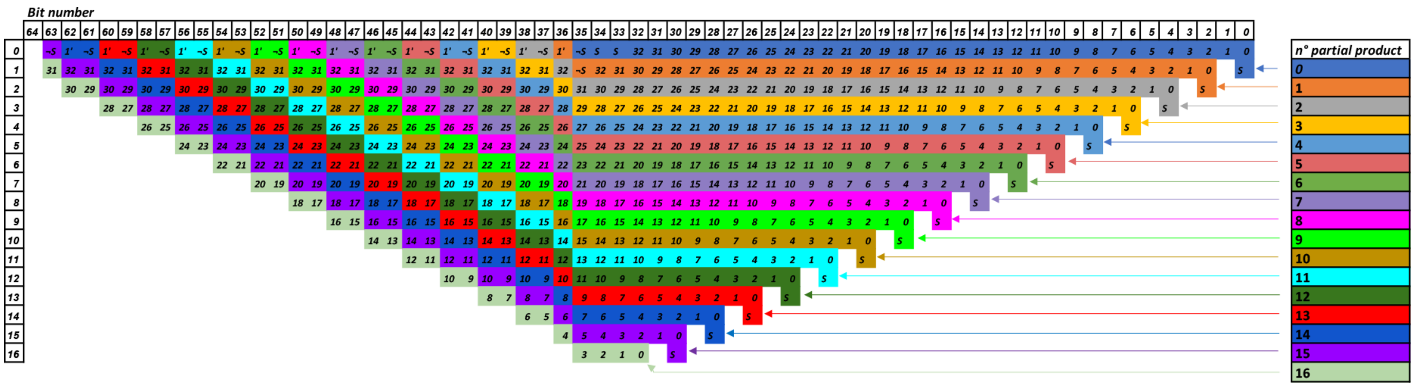
first lines. The figures from 3.1c to 3.1i are a representation of every stage in dot notation, for each figure you can see the weight of all the bits and the belonging line. The chains of FAs and HAs are highlighted using the same background color, and in the next stage the dots are depicted using the same background color of the chain from which they are originated. A black dot means that it has crossed the previous stage without any computation. In this way the connections between the inputs of a FAs (or a HAs) and a stage, and the connections between the outputs of the same FAs (or HAs) and the following stage, are immediately clear.

Once only two lines are present at the stage 0, a simple behavioral two-inputs adder is used to compute the final result of the multiplication. In this way a value of 65 bits is obtained, but the last bit represents only a carry due to the signs of the partial products (remember that they can be either positive or negative), and so it has to be discharged.

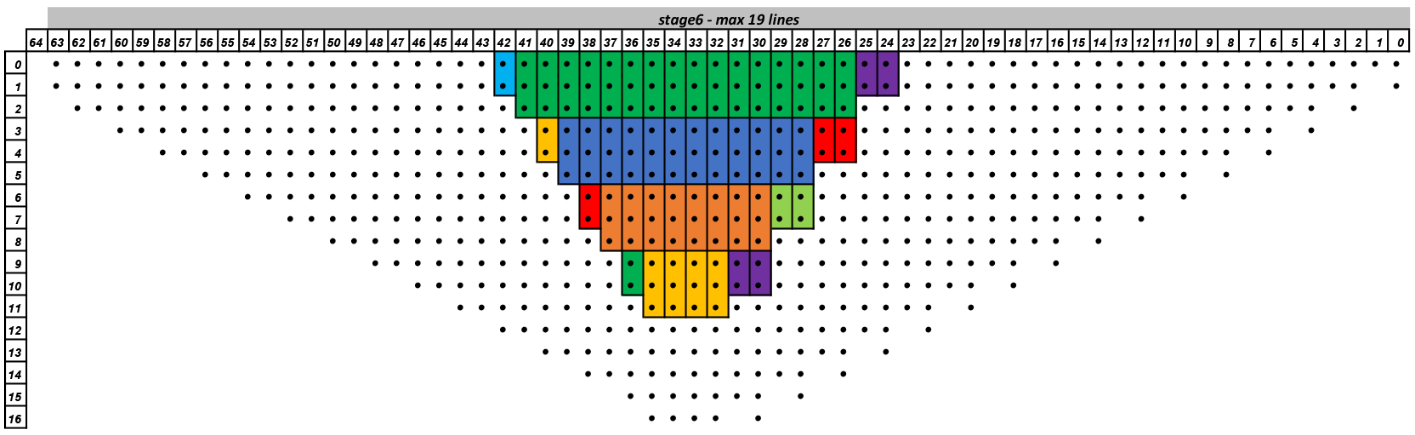
Finally, a brief simulation is performed using the testbench of listing B.7. In figure 3.2 you can see the waveforms relative to the computing of some multiplications with critical values. In figure 3.3 it is possible to see in detail the waveforms of the stage 6, the pyramidal form is the same provided by figure 3.1c.



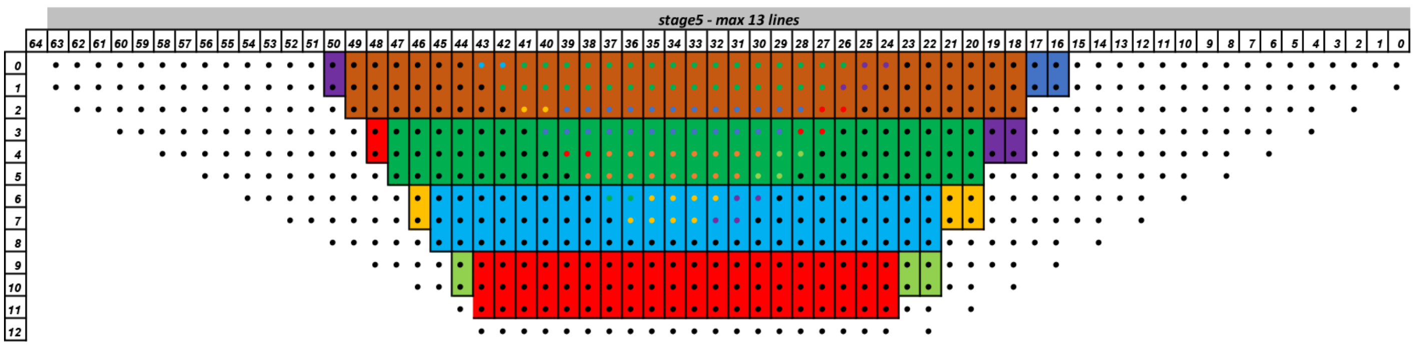
(a) *Partial products and signs generation and relative weight.*



(b) *How the partial products and their sign compose the stage 6.*

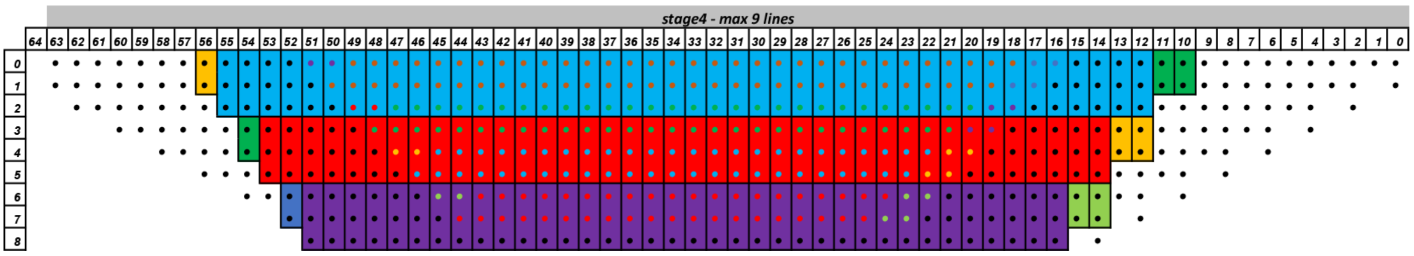


(c) *Stage 6.*

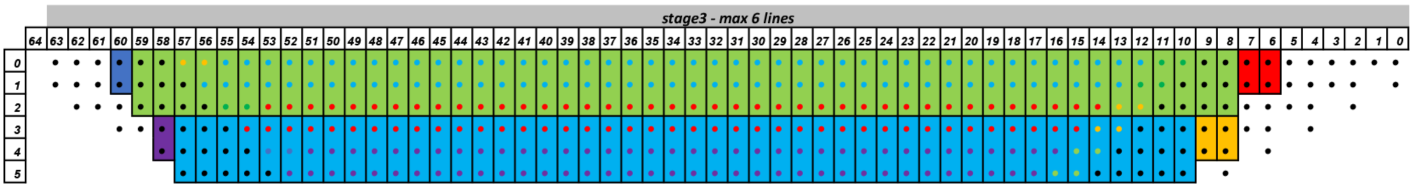


(d) *Stage 5.*

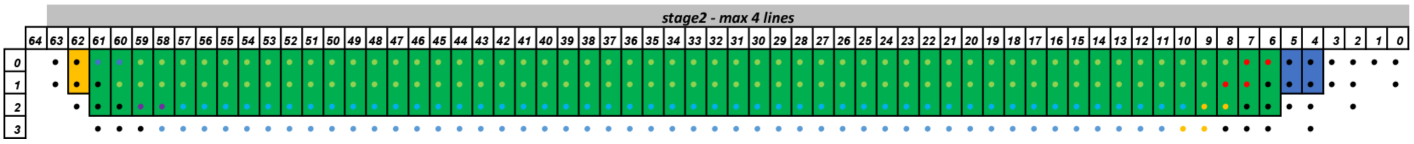
Figure 3.1: Dadda tree planning.



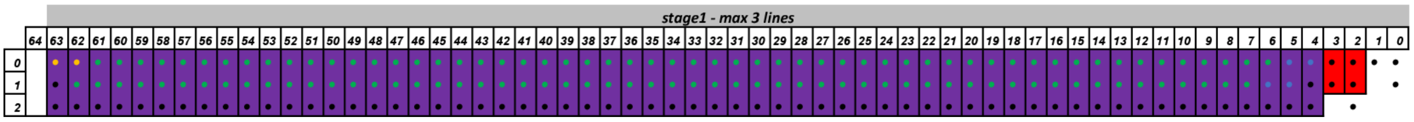
(e) *Stage 4.*



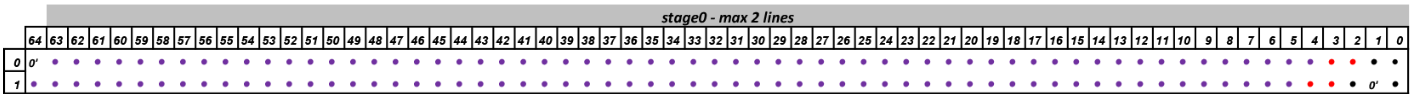
(f) *Stage 3.*



(g) *Stage 2.*

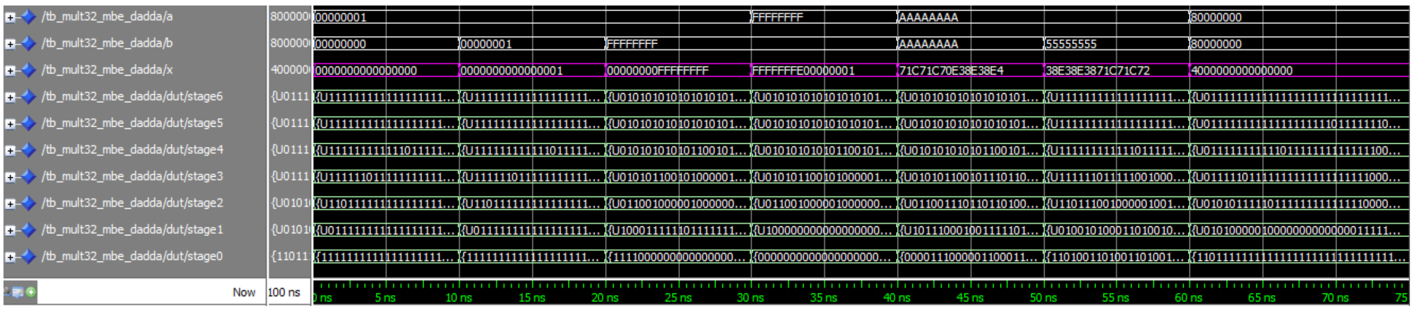


(h) *Stage 1.*



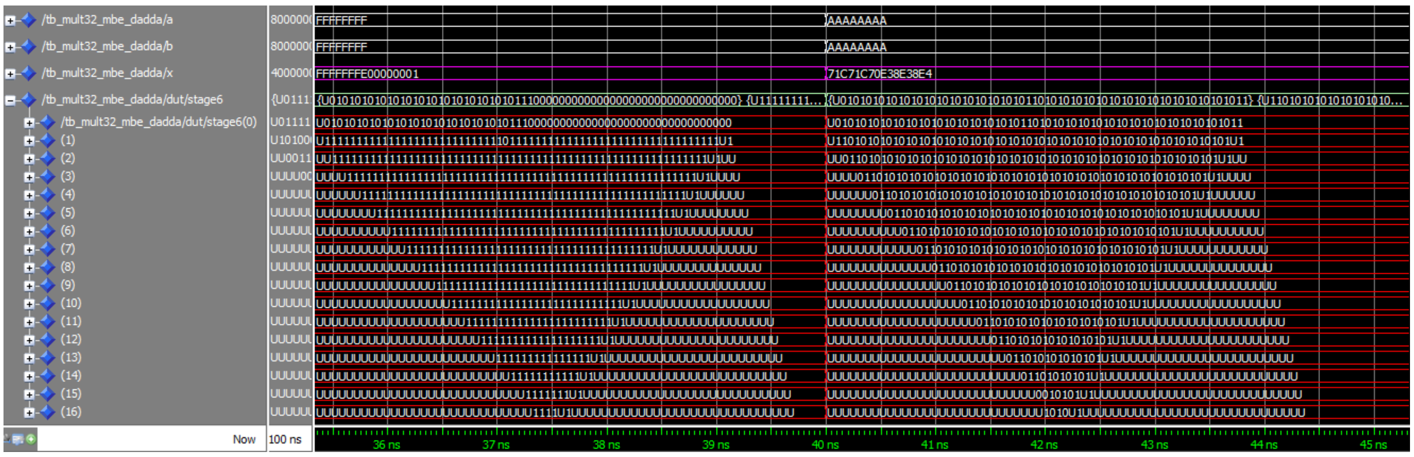
(i) *Stage 0.*

Figure 3.1: Dadda tree planning.



```
/tb_mult32_mbe_dadda/a            800000  00000001                                                                       FFFFFFFF                  AAAAAAAA                                            80000000
/tb_mult32_mbe_dadda/b            800000  00000000                 00000001                 FFFFFFFF                                            AAAAAAAA                  55555555                  80000000
/tb_mult32_mbe_dadda/x            400000  0000000000000000         0000000000000001         00000000FFFFFFFF         FFFFFFFE00000001          71C71C70E38E38E4          38E38E3871C71C72          4000000000000000
/tb_mult32_mbe_dadda/dut/stage6   {U0111  {U1111111111111111111... {U1111111111111111111... {U0101010101010101010... {U0101010101010101010... {U0101010101010101010... {U1111111111111111111... {U0111111111111111111111111111...
/tb_mult32_mbe_dadda/dut/stage5   {U0111  {U1111111111111111111... {U1111111111111111111... {U0101010101010101010... {U0101010101010101010... {U0101010101010101010... {U1111111111111111111... {U0111111111111111111101111110...
/tb_mult32_mbe_dadda/dut/stage4   {U0111  {U1111111111110111111... {U1111111111110111111... {U0101010101011001010... {U0101010101011001010... {U0101010101011001010... {U1111111111110111111... {U0111111111110111111111111100...
/tb_mult32_mbe_dadda/dut/stage3   {U0111  {U1111110111111111111... {U1111110111111111111... {U0101011001010000010... {U0101011001010000010... {U0101011001011101100... {U1111110111111001000... {U0111110111111111111111111000...
/tb_mult32_mbe_dadda/dut/stage2   {U0101  {U1101111111111111111... {U1101111111111111111... {U0110010000010000000... {U0110010000010000000... {U0110011101101101000... {U1101110010000001001... {U0101011111011111111111110000...
/tb_mult32_mbe_dadda/dut/stage1   {U0101  {U0111111111111111111... {U0111111111111111111... {U1000111111011111111... {U1000000000000000000... {U1011100010011111010... {U0100101000110100100... {U0101000001000000000000011111...
/tb_mult32_mbe_dadda/dut/stage0   {11011  {1111111111111111111...  {1111111111111111111...  {1111000000000000000...  {0000000000000000000...  {0000111000000110001...  {1101001101001101001...  {1101111111111111111111111111...
Now                               100 ns  0 ns    5 ns    10 ns    15 ns    20 ns    25 ns    30 ns    35 ns    40 ns    45 ns    50 ns    55 ns    60 ns    65 ns    70 ns    75
```

Figure 3.2: MBE multiplier simulation.



```
/tb_mult32_mbe_dadda/a                  800000( FFFFFFFF                                                                    AAAAAAAA
/tb_mult32_mbe_dadda/b                  800000( FFFFFFFF                                                                    AAAAAAAA
/tb_mult32_mbe_dadda/x                  400000( FFFFFFFE00000001                                                            71C71C70E38E38E4
/tb_mult32_mbe_dadda/dut/stage6         {U0111  {U01010101010101010101010101010101011100000000000000000000000000000000} {U11111111...   {U0101010101010101010101010101010101011010101010101010101010101010101011} {U1101010101010101010...
  /tb_mult32_mbe_dadda/dut/stage6(0)    U01111  U010101010101010101010101010101010111000000000000000000000000000000000000    U01010101010101010101010101010101010110101010101010101010101010101010101011
  (1)                                   U10100( U1111111111111111111111111111111110111111111111111111111111111111111U1        U110101010101010101010101010101010101010101010101010101010101010101U1
  (2)                                   UU0011  UU111111111111111111111111111111111111111111111111111111111111111U1UU         UU0110101010101010101010101010101010101010101010101010101010101U1UU
  (3)                                   UUUU0(  UUUU11111111111111111111111111111111111111111111111111111111U1UUUU            UUUU01101010101010101010101010101010101010101010101010101010U1UUUU
  (4)                                   UUUUUU  UUUUUU11111111111111111111111111111111111111111111111111111U1UUUUUU           UUUUUU0110101010101010101010101010101010101010101010101010U1UUUUUU
  (5)                                   UUUUUU  UUUUUUUU111111111111111111111111111111111111111111111111U1UUUUUUUU           UUUUUUUU01101010101010101010101010101010101010101010101U1UUUUUUUU
  (6)                                   UUUUUU  UUUUUUUUUU1111111111111111111111111111111111111111111U1UUUUUUUUUU            UUUUUUUUUU011010101010101010101010101010101010101010U1UUUUUUUUUU
  (7)                                   UUUUUU  UUUUUUUUUUUU11111111111111111111111111111111111111U1UUUUUUUUUUUU             UUUUUUUUUUUU0110101010101010101010101010101010101U1UUUUUUUUUUUU
  (8)                                   UUUUUU  UUUUUUUUUUUUUU111111111111111111111111111111111U1UUUUUUUUUUUUUU              UUUUUUUUUUUUUU01101010101010101010101010101010U1UUUUUUUUUUUUUU
  (9)                                   UUUUUU  UUUUUUUUUUUUUUUU1111111111111111111111111111U1UUUUUUUUUUUUUUUU               UUUUUUUUUUUUUUUU011010101010101010101010101U1UUUUUUUUUUUUUUUU
  (10)                                  UUUUUU  UUUUUUUUUUUUUUUUUU11111111111111111111111U1UUUUUUUUUUUUUUUUUU                UUUUUUUUUUUUUUUUUU0110101010101010101010U1UUUUUUUUUUUUUUUUUU
  (11)                                  UUUUUU  UUUUUUUUUUUUUUUUUUUU111111111111111111U1UUUUUUUUUUUUUUUUUUUU                 UUUUUUUUUUUUUUUUUUUU01101010101010101U1UUUUUUUUUUUUUUUUUUUU
  (12)                                  UUUUUU  UUUUUUUUUUUUUUUUUUUUUU1111111111111U1UUUUUUUUUUUUUUUUUUUUUU                  UUUUUUUUUUUUUUUUUUUUUU011010101010U1UUUUUUUUUUUUUUUUUUUUUU
  (13)                                  UUUUUU  UUUUUUUUUUUUUUUUUUUUUUUU11111111U1UUUUUUUUUUUUUUUUUUUUUUUU                   UUUUUUUUUUUUUUUUUUUUUUUU0110101U1UUUUUUUUUUUUUUUUUUUUUUUU
  (14)                                  UUUUUU  UUUUUUUUUUUUUUUUUUUUUUUUUU111U1UUUUUUUUUUUUUUUUUUUUUUUUUU                    UUUUUUUUUUUUUUUUUUUUUUUUUU0110101U1UUUUUUUUUUUUUUUUUUUUUUUUUU
  (15)                                  UUUUUU  UUUUUUUUUUUUUUUUUUUUUUUUUUUU1111111U1UUUUUUUUUUUUUUUUUUUUUUUUUUUU             UUUUUUUUUUUUUUUUUUUUUUUUUUUU0010101U1UUUUUUUUUUUUUUUUUUUUUUUUUUUU
  (16)                                  UUUUUU  UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU1111U1UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU            UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU1010U1UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
Now                                     100 ns    36 ns   37 ns   38 ns   39 ns   40 ns   41 ns   42 ns   43 ns   44 ns   45 ns
```

Figure 3.3: Detail of the stage 6 during the simulation.

APPENDIX A

VHDL listings

Listing A.1: fpmul_pipeline_reg_in.vhd

```
-- VHDL Entity HAVOC.FPmul.symbol
--
-- Created by
-- Guillermo Marcus, gmarcus@ieee.org
-- using Mentor Graphics FPGA Advantage tools.
--
-- Visit "http://fpga.mty.itesm.mx" for more info.
--
-- 2003-2004. V1.0
--

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

ENTITY FPmul IS
   PORT(
      FP_A : IN     std_logic_vector (31 DOWNTO 0);
      FP_B : IN     std_logic_vector (31 DOWNTO 0);
      clk  : IN     std_logic;
      FP_Z : OUT    std_logic_vector (31 DOWNTO 0));
-- Declarations
END FPmul ;

--
-- VHDL Architecture HAVOC.FPmul.pipeline
--
-- Created by
-- Guillermo Marcus, gmarcus@ieee.org
-- using Mentor Graphics FPGA Advantage tools.
--
-- Visit "http://fpga.mty.itesm.mx" for more info.
--
-- Copyright 2003-2004. V1.0
--

ARCHITECTURE pipeline OF FPmul IS

   -- Architecture declarations

   -- Internal signal declarations
    signal A_reg            : std_logic_vector(31 DOWNTO 0);
    signal B_reg            : std_logic_vector(31 DOWNTO 0);

   SIGNAL A_EXP           : std_logic_vector(7 DOWNTO 0);
   SIGNAL A_SIG           : std_logic_vector(31 DOWNTO 0);
   SIGNAL B_EXP           : std_logic_vector(7 DOWNTO 0);
   SIGNAL B_SIG           : std_logic_vector(31 DOWNTO 0);
   SIGNAL EXP_in          : std_logic_vector(7 DOWNTO 0);
   SIGNAL EXP_neg         : std_logic;
   SIGNAL EXP_neg_stage2  : std_logic;
   SIGNAL EXP_out_round   : std_logic_vector(7 DOWNTO 0);
   SIGNAL EXP_pos         : std_logic;
   SIGNAL EXP_pos_stage2  : std_logic;
   SIGNAL SIGN_out        : std_logic;
   SIGNAL SIGN_out_stage1 : std_logic;
   SIGNAL SIGN_out_stage2 : std_logic;
   SIGNAL SIG_in          : std_logic_vector(27 DOWNTO 0);
   SIGNAL SIG_out_round   : std_logic_vector(27 DOWNTO 0);
   SIGNAL isINF_stage1    : std_logic;
   SIGNAL isINF_stage2    : std_logic;
   SIGNAL isINF_tab       : std_logic;
   SIGNAL isNaN           : std_logic;
   SIGNAL isNaN_stage1    : std_logic;
   SIGNAL isNaN_stage2    : std_logic;
   SIGNAL isZ_tab         : std_logic;
   SIGNAL isZ_tab_stage1  : std_logic;
   SIGNAL isZ_tab_stage2  : std_logic;


   -- Component Declarations
   COMPONENT reg is
    Generic (N: positive:= 1 );                            --  number of bits
    Port(   D      : In    std_logic_vector(N-1 downto 0);      --  data input
            Q      : Out   std_logic_vector(N-1 downto 0);      --  data output
            CLK    : In    std_logic                            -- clock signal
    );
    end COMPONENT reg;

   COMPONENT FPmul_stage1
   PORT (
      FP_A            : IN     std_logic_vector (31 DOWNTO 0);
      FP_B            : IN     std_logic_vector (31 DOWNTO 0);
      clk             : IN     std_logic ;
      A_EXP           : OUT    std_logic_vector (7 DOWNTO 0);
      A_SIG           : OUT    std_logic_vector (31 DOWNTO 0);
      B_EXP           : OUT    std_logic_vector (7 DOWNTO 0);
      B_SIG           : OUT    std_logic_vector (31 DOWNTO 0);
      SIGN_out_stage1 : OUT    std_logic ;
      isINF_stage1    : OUT    std_logic ;
      isNaN_stage1    : OUT    std_logic ;
      isZ_tab_stage1  : OUT    std_logic
   );
   END COMPONENT;
   COMPONENT FPmul_stage2
   PORT (
      A_EXP           : IN     std_logic_vector (7 DOWNTO 0);
      A_SIG           : IN     std_logic_vector (31 DOWNTO 0);
      B_EXP           : IN     std_logic_vector (7 DOWNTO 0);
      B_SIG           : IN     std_logic_vector (31 DOWNTO 0);
      SIGN_out_stage1 : IN     std_logic ;
      clk             : IN     std_logic ;
      isINF_stage1    : IN     std_logic ;
      isNaN_stage1    : IN     std_logic ;
      isZ_tab_stage1  : IN     std_logic ;
      EXP_in          : OUT    std_logic_vector (7 DOWNTO 0);
      EXP_neg_stage2  : OUT    std_logic ;
      EXP_pos_stage2  : OUT    std_logic ;
      SIGN_out_stage2 : OUT    std_logic ;
      SIG_in          : OUT    std_logic_vector (27 DOWNTO 0);
      isINF_stage2    : OUT    std_logic ;
      isNaN_stage2    : OUT    std_logic ;
      isZ_tab_stage2  : OUT    std_logic
   );
   END COMPONENT;
   COMPONENT FPmul_stage3
   PORT (
      EXP_in          : IN     std_logic_vector (7 DOWNTO 0);
      EXP_neg_stage2  : IN     std_logic ;
```

```
      EXP_pos_stage2  : IN     std_logic ;
      SIGN_out_stage2 : IN     std_logic ;
      SIG_in          : IN     std_logic_vector (27 DOWNTO 0);
      clk             : IN     std_logic ;
      isINF_stage2    : IN     std_logic ;
      isNaN_stage2    : IN     std_logic ;
      isZ_tab_stage2  : IN     std_logic ;
      EXP_neg         : OUT    std_logic ;
      EXP_out_round   : OUT    std_logic_vector (7 DOWNTO 0);
      EXP_pos         : OUT    std_logic ;
      SIGN_out        : OUT    std_logic ;
      SIG_out_round   : OUT    std_logic_vector (27 DOWNTO 0);
      isINF_tab       : OUT    std_logic ;
      isNaN           : OUT    std_logic ;
      isZ_tab         : OUT    std_logic
   );
   END COMPONENT;
   COMPONENT FPmul_stage4
   PORT (
      EXP_neg       : IN     std_logic ;
      EXP_out_round : IN     std_logic_vector (7 DOWNTO 0);
      EXP_pos       : IN     std_logic ;
      SIGN_out      : IN     std_logic ;
      SIG_out_round : IN     std_logic_vector (27 DOWNTO 0);
      clk           : IN     std_logic ;
      isINF_tab     : IN     std_logic ;
      isNaN         : IN     std_logic ;
      isZ_tab       : IN     std_logic ;
      FP_Z          : OUT    std_logic_vector (31 DOWNTO 0)
   );
   END COMPONENT;

BEGIN

   -- Instance port mappings.
   -- internal register used for the inputs
    register_A: reg
    Generic map(N=>32 )
    Port map(   D   => FP_A,
                Q   => A_reg,
                CLK => clk);
    register_B: reg
    Generic map(N=>32 )
    Port map(   D   => FP_B,
                Q   => B_reg,
                CLK => clk);
    -------------------------------------------

   I1 : FPmul_stage1
      PORT MAP (
         FP_A            => A_reg,
         FP_B            => B_reg,
         clk             => clk,
         A_EXP           => A_EXP,
         A_SIG           => A_SIG,
         B_EXP           => B_EXP,
         B_SIG           => B_SIG,
         SIGN_out_stage1 => SIGN_out_stage1,
         isINF_stage1    => isINF_stage1,
         isNaN_stage1    => isNaN_stage1,
         isZ_tab_stage1  => isZ_tab_stage1
      );
   I2 : FPmul_stage2
      PORT MAP (
         A_EXP           => A_EXP,
         A_SIG           => A_SIG,
         B_EXP           => B_EXP,
         B_SIG           => B_SIG,
         SIGN_out_stage1 => SIGN_out_stage1,
         clk             => clk,
         isINF_stage1    => isINF_stage1,
         isNaN_stage1    => isNaN_stage1,
         isZ_tab_stage1  => isZ_tab_stage1,
         EXP_in          => EXP_in,
         EXP_neg_stage2  => EXP_neg_stage2,
         EXP_pos_stage2  => EXP_pos_stage2,
         SIGN_out_stage2 => SIGN_out_stage2,
         SIG_in          => SIG_in,
         isINF_stage2    => isINF_stage2,
         isNaN_stage2    => isNaN_stage2,
         isZ_tab_stage2  => isZ_tab_stage2
      );
   I3 : FPmul_stage3
      PORT MAP (
         EXP_in          => EXP_in,
         EXP_neg_stage2  => EXP_neg_stage2,
         EXP_pos_stage2  => EXP_pos_stage2,
         SIGN_out_stage2 => SIGN_out_stage2,
         SIG_in          => SIG_in,
         clk             => clk,
         isINF_stage2    => isINF_stage2,
         isNaN_stage2    => isNaN_stage2,
         isZ_tab_stage2  => isZ_tab_stage2,
         EXP_neg         => EXP_neg,
         EXP_out_round   => EXP_out_round,
         EXP_pos         => EXP_pos,
         SIGN_out        => SIGN_out,
         SIG_out_round   => SIG_out_round,
         isINF_tab       => isINF_tab,
         isNaN           => isNaN,
         isZ_tab         => isZ_tab
      );
   I4 : FPmul_stage4
      PORT MAP (
         EXP_neg       => EXP_neg,
         EXP_out_round => EXP_out_round,
         EXP_pos       => EXP_pos,
         SIGN_out      => SIGN_out,
         SIG_out_round => SIG_out_round,
         clk           => clk,
         isINF_tab     => isINF_tab,
         isNaN         => isNaN,
         isZ_tab       => isZ_tab,
         FP_Z          => FP_Z
      );

END pipeline;
```

Listing A.2: fpmul_pipeline_additional_reg.vhd

```
-- VHDL Entity HAVOC.FPmul.symbol
--
-- Created by
-- Guillermo Marcus, gmarcus@ieee.org
-- using Mentor Graphics FPGA Advantage tools.
--
-- Visit "http://fpga.mty.itesm.mx" for more info.
--
-- 2003-2004. V1.0
--

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

ENTITY FPmul IS
   PORT(
      FP_A : IN     std_logic_vector (31 DOWNTO 0);
      FP_B : IN     std_logic_vector (31 DOWNTO 0);
      clk  : IN     std_logic;
      FP_Z : OUT    std_logic_vector (31 DOWNTO 0));
```

```
-- Declarations
END FPmul ;

--
-- VHDL Architecture HAVOC.FPmul.pipeline
--
-- Created by
-- Guillermo Marcus, gmarcus@ieee.org
-- using Mentor Graphics FPGA Advantage tools.
--
-- Visit "http://fpga.mty.itesm.mx" for more info.
--
-- Copyright 2003-2004. V1.0
--

ARCHITECTURE pipeline OF FPmul IS

   -- Architecture declarations

   -- Internal signal declarations
    signal A_reg            : std_logic_vector(31 DOWNTO 0);
    signal B_reg            : std_logic_vector(31 DOWNTO 0);

   SIGNAL A_EXP           : std_logic_vector(7 DOWNTO 0);
   SIGNAL A_SIG           : std_logic_vector(31 DOWNTO 0);
   SIGNAL B_EXP           : std_logic_vector(7 DOWNTO 0);
   SIGNAL B_SIG           : std_logic_vector(31 DOWNTO 0);
   SIGNAL EXP_in          : std_logic_vector(7 DOWNTO 0);
   SIGNAL EXP_neg         : std_logic;
   SIGNAL EXP_neg_stage2  : std_logic;
   SIGNAL EXP_out_round   : std_logic_vector(7 DOWNTO 0);
   SIGNAL EXP_pos         : std_logic;
   SIGNAL EXP_pos_stage2  : std_logic;
   SIGNAL SIGN_out        : std_logic;
   SIGNAL SIGN_out_stage1 : std_logic;
   SIGNAL SIGN_out_stage2 : std_logic;
   SIGNAL SIG_in          : std_logic_vector(27 DOWNTO 0);
   SIGNAL SIG_out_round   : std_logic_vector(27 DOWNTO 0);
   SIGNAL isINF_stage1    : std_logic;
   SIGNAL isINF_stage2    : std_logic;
   SIGNAL isINF_tab       : std_logic;
   SIGNAL isNaN           : std_logic;
   SIGNAL isNaN_stage1    : std_logic;
   SIGNAL isNaN_stage2    : std_logic;
   SIGNAL isZ_tab         : std_logic;
   SIGNAL isZ_tab_stage1  : std_logic;
   SIGNAL isZ_tab_stage2  : std_logic;
   -- added by ISA25:
       ↪ ---------------------------------------------------------------------------
   SIGNAL EXP_in_reg         : std_logic_vector(7 DOWNTO 0);
   SIGNAL EXP_neg_stage2_reg  : std_logic;
   SIGNAL EXP_pos_stage2_reg  : std_logic;
   SIGNAL SIGN_out_stage2_reg : std_logic;
   SIGNAL SIG_in_reg          : std_logic_vector(27 DOWNTO 0);
   SIGNAL isINF_stage2_reg    : std_logic;
   SIGNAL isNaN_stage2_reg    : std_logic;
   SIGNAL isZ_tab_stage2_reg  : std_logic;


   -- Component Declarations

   COMPONENT reg is
    Generic (N: positive:= 1 );                                 --  number of bits
    Port(   D       : In    std_logic_vector(N-1 downto 0);      --  data input
            Q       : Out   std_logic_vector(N-1 downto 0);      --  data output
            CLK     : In    std_logic                            -- clock signal
    );
    end COMPONENT reg;

   COMPONENT FPmul_stage1
   PORT (
      FP_A            : IN     std_logic_vector (31 DOWNTO 0);
      FP_B            : IN     std_logic_vector (31 DOWNTO 0);
      clk             : IN     std_logic ;
      A_EXP           : OUT    std_logic_vector (7 DOWNTO 0);
      A_SIG           : OUT    std_logic_vector (31 DOWNTO 0);
      B_EXP           : OUT    std_logic_vector (7 DOWNTO 0);
      B_SIG           : OUT    std_logic_vector (31 DOWNTO 0);
      SIGN_out_stage1 : OUT    std_logic ;
      isINF_stage1    : OUT    std_logic ;
      isNaN_stage1    : OUT    std_logic ;
      isZ_tab_stage1  : OUT    std_logic
   );
   END COMPONENT;
   COMPONENT FPmul_stage2
   PORT (
      A_EXP           : IN     std_logic_vector (7 DOWNTO 0);
      A_SIG           : IN     std_logic_vector (31 DOWNTO 0);
      B_EXP           : IN     std_logic_vector (7 DOWNTO 0);
      B_SIG           : IN     std_logic_vector (31 DOWNTO 0);
      SIGN_out_stage1 : IN     std_logic ;
      clk             : IN     std_logic ;
      isINF_stage1    : IN     std_logic ;
      isNaN_stage1    : IN     std_logic ;
      isZ_tab_stage1  : IN     std_logic ;
      EXP_in          : OUT    std_logic_vector (7 DOWNTO 0);
      EXP_neg_stage2  : OUT    std_logic ;
      EXP_pos_stage2  : OUT    std_logic ;
      SIGN_out_stage2 : OUT    std_logic ;
      SIG_in          : OUT    std_logic_vector (27 DOWNTO 0);
      isINF_stage2    : OUT    std_logic ;
      isNaN_stage2    : OUT    std_logic ;
      isZ_tab_stage2  : OUT    std_logic
   );
   END COMPONENT;
   COMPONENT FPmul_stage3
   PORT (
      EXP_in          : IN     std_logic_vector (7 DOWNTO 0);
      EXP_neg_stage2  : IN     std_logic ;
      EXP_pos_stage2  : IN     std_logic ;
      SIGN_out_stage2 : IN     std_logic ;
      SIG_in          : IN     std_logic_vector (27 DOWNTO 0);
      clk             : IN     std_logic ;
      isINF_stage2    : IN     std_logic ;
      isNaN_stage2    : IN     std_logic ;
      isZ_tab_stage2  : IN     std_logic ;
      EXP_neg         : OUT    std_logic ;
      EXP_out_round   : OUT    std_logic_vector (7 DOWNTO 0);
      EXP_pos         : OUT    std_logic ;
      SIGN_out        : OUT    std_logic ;
      SIG_out_round   : OUT    std_logic_vector (27 DOWNTO 0);
      isINF_tab       : OUT    std_logic ;
      isNaN           : OUT    std_logic ;
      isZ_tab         : OUT    std_logic
   );
   END COMPONENT;
   COMPONENT FPmul_stage4
   PORT (
      EXP_neg       : IN     std_logic ;
      EXP_out_round : IN     std_logic_vector (7 DOWNTO 0);
      EXP_pos       : IN     std_logic ;
      SIGN_out      : IN     std_logic ;
      SIG_out_round : IN     std_logic_vector (27 DOWNTO 0);
      clk           : IN     std_logic ;
      isINF_tab     : IN     std_logic ;
      isNaN         : IN     std_logic ;
      isZ_tab       : IN     std_logic ;
      FP_Z          : OUT    std_logic_vector (31 DOWNTO 0)
   );
   END COMPONENT;

BEGIN

   -- Instance port mappings.
   -- internal register used for the inputs
    register_A: reg
    Generic map(N=>32 )
    Port map(   D   => FP_A,
                Q   => A_reg,
                CLK => clk);
    register_B: reg
    Generic map(N=>32 )
    Port map(   D   => FP_B,
                Q   => B_reg,
                CLK => clk);
    -------------------------------------------
    -- registers at the output of the Significand Multiplier on stage 2

    register_EXP_in: reg
    Generic map(N=>8 )
    Port map(   D   => EXP_in,
                Q   => EXP_in_reg,
                CLK => clk);
    register_EXP_neg_stage2: reg
    Generic map(N=>1 )
    Port map(   D(0)    => EXP_neg_stage2,
                Q(0)    => EXP_neg_stage2_reg,
                CLK => clk);
    register_EXP_pos_stage2: reg
    Generic map(N=>1 )
    Port map(   D(0)    => EXP_pos_stage2,
                Q(0)    => EXP_pos_stage2_reg,
                CLK => clk);
    register_SIGN_out_stage2 : reg
    Generic map(N=>1 )
    Port map(   D(0)    => SIGN_out_stage2,
                Q(0)    => SIGN_out_stage2_reg,
                CLK => clk);
    register_SIG_in : reg
    Generic map(N=>28 )
    Port map(   D   => SIG_in,
                Q   => SIG_in_reg,
                CLK => clk);
    register_isINF_stage2 : reg
    Generic map(N=>1 )
    Port map(   D(0)    => isINF_stage2,
                Q(0)    => isINF_stage2_reg,
                CLK => clk);
    register_isNaN_stage2 : reg
    Generic map(N=>1 )
    Port map(   D(0)    => isNaN_stage2,
                Q(0)    => isNaN_stage2_reg,
                CLK => clk);
    register_isZ_tab_stage2 : reg
    Generic map(N=>1 )
    Port map(   D(0)    => isZ_tab_stage2,
                Q(0)    => isZ_tab_stage2_reg,
                CLK => clk);


   I1 : FPmul_stage1
      PORT MAP (
         FP_A            => A_reg,
         FP_B            => B_reg,
         clk             => clk,
         A_EXP           => A_EXP,
         A_SIG           => A_SIG,
         B_EXP           => B_EXP,
         B_SIG           => B_SIG,
         SIGN_out_stage1 => SIGN_out_stage1,
         isINF_stage1    => isINF_stage1,
         isNaN_stage1    => isNaN_stage1,
         isZ_tab_stage1  => isZ_tab_stage1
      );
   I2 : FPmul_stage2
      PORT MAP (
         A_EXP           => A_EXP,
         A_SIG           => A_SIG,
         B_EXP           => B_EXP,
         B_SIG           => B_SIG,
         SIGN_out_stage1 => SIGN_out_stage1,
         clk             => clk,
         isINF_stage1    => isINF_stage1,
         isNaN_stage1    => isNaN_stage1,
         isZ_tab_stage1  => isZ_tab_stage1,
         EXP_in          => EXP_in,
         EXP_neg_stage2  => EXP_neg_stage2,
         EXP_pos_stage2  => EXP_pos_stage2,
         SIGN_out_stage2 => SIGN_out_stage2,
         SIG_in          => SIG_in,
         isINF_stage2    => isINF_stage2,
         isNaN_stage2    => isNaN_stage2,
         isZ_tab_stage2  => isZ_tab_stage2
      );
   I3 : FPmul_stage3
      PORT MAP (
         EXP_in          => EXP_in_reg,
         EXP_neg_stage2  => EXP_neg_stage2_reg,
         EXP_pos_stage2  => EXP_pos_stage2_reg,
         SIGN_out_stage2 => SIGN_out_stage2_reg,
         SIG_in          => SIG_in_reg,
         clk             => clk,
         isINF_stage2    => isINF_stage2_reg,
         isNaN_stage2    => isNaN_stage2_reg,
         isZ_tab_stage2  => isZ_tab_stage2_reg,
         EXP_neg         => EXP_neg,
         EXP_out_round   => EXP_out_round,
         EXP_pos         => EXP_pos,
         SIGN_out        => SIGN_out,
         SIG_out_round   => SIG_out_round,
         isINF_tab       => isINF_tab,
         isNaN           => isNaN,
         isZ_tab         => isZ_tab
      );
   I4 : FPmul_stage4
      PORT MAP (
         EXP_neg       => EXP_neg,
         EXP_out_round => EXP_out_round,
         EXP_pos       => EXP_pos,
         SIGN_out      => SIGN_out,
         SIG_out_round => SIG_out_round,
         clk           => clk,
         isINF_tab     => isINF_tab,
         isNaN         => isNaN,
         isZ_tab       => isZ_tab,
         FP_Z          => FP_Z
      );

END pipeline;
```

Listing A.3: fpmul_stage2_struct_MBE.vhd

```
-- VHDL Entity HAVOC.FPmul_stage2.interface
--
-- Created by
-- Guillermo Marcus, gmarcus@ieee.org
-- using Mentor Graphics FPGA Advantage tools.
--
-- Visit "http://fpga.mty.itesm.mx" for more info.
--
-- 2003-2004. V1.0
--

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
use work.pack_mult32_MBE_dadda.all;

ENTITY FPmul_stage2 IS
   PORT(
      A_EXP           : IN     std_logic_vector (7 DOWNTO 0);
      A_SIG           : IN     std_logic_vector (31 DOWNTO 0);
      B_EXP           : IN     std_logic_vector (7 DOWNTO 0);
      B_SIG           : IN     std_logic_vector (31 DOWNTO 0);
      SIGN_out_stage1 : IN     std_logic;
      clk             : IN     std_logic;
      isINF_stage1    : IN     std_logic;
      isNaN_stage1    : IN     std_logic;
      isZ_tab_stage1  : IN     std_logic;
      EXP_in          : OUT    std_logic_vector (7 DOWNTO 0);
      EXP_neg_stage2  : OUT    std_logic;
      EXP_pos_stage2  : OUT    std_logic;
      SIGN_out_stage2 : OUT    std_logic;
      SIG_in          : OUT    std_logic_vector (27 DOWNTO 0);
      isINF_stage2    : OUT    std_logic;
      isNaN_stage2    : OUT    std_logic;
      isZ_tab_stage2  : OUT    std_logic
   );

-- Declarations

END FPmul_stage2 ;

--
-- VHDL Architecture HAVOC.FPmul_stage2.struct
--
-- Created by
-- Guillermo Marcus, gmarcus@ieee.org
-- using Mentor Graphics FPGA Advantage tools.
--
-- Visit "http://fpga.mty.itesm.mx" for more info.
--
-- Copyright 2003-2004. V1.0
--

ARCHITECTURE struct OF FPmul_stage2 IS

   -- Architecture declarations

   -- Internal signal declarations
   SIGNAL EXP_in_int  : std_logic_vector(7 DOWNTO 0);
   SIGNAL EXP_neg_int : std_logic;
   SIGNAL EXP_pos_int : std_logic;
   SIGNAL SIG_in_int  : std_logic_vector(27 DOWNTO 0);
   SIGNAL dout        : std_logic;
   SIGNAL dout1       : std_logic_vector(7 DOWNTO 0);
   SIGNAL prod        : std_logic_vector(63 DOWNTO 0);

COMPONENT mult32_MBE_dadda is
    port(
        a, b: in std_logic_vector(N-1 downto 0);
        x: out std_logic_vector(2*N-1 downto 0)
    );
end COMPONENT mult32_MBE_dadda;

BEGIN
   -- Architecture concurrent statements
   -- HDL Embedded Text Block 1 sig
   -- eb1 1
   SIG_in_int <= prod(47 DOWNTO 20);

   -- HDL Embedded Text Block 2 inv
   -- eb5 5
   EXP_in_int <= (NOT dout1(7)) & dout1(6 DOWNTO 0);

   -- HDL Embedded Text Block 3 latch
   -- eb2 2
   PROCESS(clk)
   BEGIN
      IF RISING_EDGE(clk) THEN
         EXP_in <= EXP_in_int;
         SIG_in <= SIG_in_int;
         EXP_pos_stage2 <= EXP_pos_int;
         EXP_neg_stage2 <= EXP_neg_int;
      END IF;
   END PROCESS;

   -- HDL Embedded Text Block 4 latch2
   -- latch2 4
   PROCESS(clk)
   BEGIN
      IF RISING_EDGE(clk) THEN
         isINF_stage2 <= isINF_stage1;
         isNaN_stage2 <= isNaN_stage1;
         isZ_tab_stage2 <= isZ_tab_stage1;
         SIGN_out_stage2 <= SIGN_out_stage1;
      END IF;
   END PROCESS;

   -- HDL Embedded Text Block 5 eb1
   -- exp_pos 5
   EXP_pos_int <= A_EXP(7) AND B_EXP(7);
--   EXP_neg_int <= NOT(A_EXP(7) OR B_EXP(7));
   EXP_neg_int <= '1' WHEN ( (A_EXP(7)='0' AND NOT (A_EXP=X"7F")) AND (B_EXP(7)='0' AND NOT (
       ↪ B_EXP=X"7F")) ) ELSE '0';


   -- ModuleWare code(v1.1) for instance 'I4' of 'add'
   I4combo: PROCESS (A_EXP, B_EXP, dout)
   VARIABLE mw_I4t0 : std_logic_vector(8 DOWNTO 0);
   VARIABLE mw_I4t1 : std_logic_vector(8 DOWNTO 0);
   VARIABLE mw_I4sum : unsigned(8 DOWNTO 0);
   VARIABLE mw_I4carry : std_logic;
   BEGIN
      mw_I4t0 := '0' & A_EXP;
      mw_I4t1 := '0' & B_EXP;
      mw_I4carry := dout;
      mw_I4sum := unsigned(mw_I4t0) + unsigned(mw_I4t1) + mw_I4carry;
      dout1 <= conv_std_logic_vector(mw_I4sum(7 DOWNTO 0),8);
   END PROCESS I4combo;

   -- Instance of the MBE multiplier with adder plane based on DADDA-tree

   MBE_mult32: mult32_MBE_dadda
    port map(
        a => A_SIG,
        b => B_SIG,
        x => prod );

   -- ModuleWare code(v1.1) for instance 'I6' of 'vdd'
   dout <= '1';

END struct;
```

Listing A.4: mult32_MBE_dadda.vhd

```
library ieee;
library work;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.pack_mult32_MBE_dadda.all;

entity mult32_MBE_dadda is
    port(
        a, b: in std_logic_vector(N-1 downto 0);
        x: out std_logic_vector(2*N-1 downto 0)
    );
end entity mult32_MBE_dadda;

architecture behavioral of mult32_MBE_dadda is

    signal b_ext: std_logic_vector_intrange(N+1 downto -1);

    signal partial_products: partial_products_type(0 to N/2);

    signal x_tmp : std_logic_vector(2*N downto 0);

    signal stage6: stage_type(0 to 16);
    signal stage5: stage_type(0 to 12);
    signal stage4: stage_type(0 to 8);
    signal stage3: stage_type(0 to 5);
    signal stage2: stage_type(0 to 3);
    signal stage1: stage_type(0 to 2);
    signal stage0: stage_type(0 to 1);

begin

    ----------------------------------------------------MULTIPLIER BITS EXTENSION
        ↪ ----------------------------------------------------

    b_ext(-1)<='0';                            --connect bit -1 of B_ext to zero
    b_ext_generate: for i in 0 to N-1 generate  --connect all other bits to input B
        b_ext(i)<=B(i);
    end generate b_ext_generate;
    b_ext(N+1 downto N)<="00";                 --connect bits 32 and 33 to zero

    -----------------------------------------------------PARTIAL PRODUCTS GENERATION
        ↪ ---------------------------------------------------

    partial_product_generate: for i in 0 to N/2 generate
        partial_products(i)<=get_MBE_partial_product(a, b_ext, i*2);
    end generate partial_product_generate;

    -----------------------------------------------------FROM PARTIAL PRODUCTS TO PYRAMIDAL FORM
        ↪ -------------------------------------------
    --stage6 line0
    stage6(0)(32 downto 0)  <=partial_products(0).value;
    stage6(0)(33)           <=partial_products(0).sign;
    stage6(0)(34)           <=partial_products(0).sign;
    stage6(0)(35)           <=not partial_products(0).sign;
    stage6(0)(36)           <='1';
    stage6(0)(37)           <=not partial_products(2).sign;
    stage6(0)(38)           <='1';
    stage6(0)(39)           <=not partial_products(3).sign;
    stage6(0)(40)           <='1';
    stage6(0)(41)           <=not partial_products(4).sign;
    stage6(0)(42)           <='1';
    stage6(0)(43)           <=not partial_products(5).sign;
    stage6(0)(44)           <='1';
    stage6(0)(45)           <=not partial_products(6).sign;
    stage6(0)(46)           <='1';
    stage6(0)(47)           <=not partial_products(7).sign;
    stage6(0)(48)           <='1';
    stage6(0)(49)           <=not partial_products(8).sign;
    stage6(0)(50)           <='1';
    stage6(0)(51)           <=not partial_products(9).sign;
    stage6(0)(52)           <='1';
    stage6(0)(53)           <=not partial_products(10).sign;
    stage6(0)(54)           <='1';
    stage6(0)(55)           <=not partial_products(11).sign;
    stage6(0)(56)           <='1';
    stage6(0)(57)           <=not partial_products(12).sign;
    stage6(0)(58)           <='1';
    stage6(0)(59)           <=not partial_products(13).sign;
    stage6(0)(60)           <='1';
    stage6(0)(61)           <=not partial_products(14).sign;
    stage6(0)(62)           <='1';
```

```
stage6(0)(63)           <=not partial_products(15).sign;
--stage6 line1
stage6(1)(0)            <=partial_products(0).sign;
stage6(1)(34 downto 2)  <=partial_products(1).value;
stage6(1)(35)           <=not partial_products(1).sign;
stage6(1)(36)           <=partial_products(2).value(32);
stage6(1)(38 downto 37) <=partial_products(3).value(32 downto 31);
stage6(1)(40 downto 39) <=partial_products(4).value(32 downto 31);
stage6(1)(42 downto 41) <=partial_products(5).value(32 downto 31);
stage6(1)(44 downto 43) <=partial_products(6).value(32 downto 31);
stage6(1)(46 downto 45) <=partial_products(7).value(32 downto 31);
stage6(1)(48 downto 47) <=partial_products(8).value(32 downto 31);
stage6(1)(50 downto 49) <=partial_products(9).value(32 downto 31);
stage6(1)(52 downto 51) <=partial_products(10).value(32 downto 31);
stage6(1)(54 downto 53) <=partial_products(11).value(32 downto 31);
stage6(1)(56 downto 55) <=partial_products(12).value(32 downto 31);
stage6(1)(58 downto 57) <=partial_products(13).value(32 downto 31);
stage6(1)(60 downto 59) <=partial_products(14).value(32 downto 31);
stage6(1)(62 downto 61) <=partial_products(15).value(32 downto 31);
stage6(1)(63)           <=partial_products(16).value(31);
--stage6 line2
stage6(2)(2)            <=partial_products(1).sign;
stage6(2)(35 downto 4)  <=partial_products(2).value(31 downto 0);
stage6(2)(36)           <=partial_products(3).value(30);
stage6(2)(38 downto 37) <=partial_products(4).value(30 downto 29);
stage6(2)(40 downto 39) <=partial_products(5).value(30 downto 29);
stage6(2)(42 downto 41) <=partial_products(6).value(30 downto 29);
stage6(2)(44 downto 43) <=partial_products(7).value(30 downto 29);
stage6(2)(46 downto 45) <=partial_products(8).value(30 downto 29);
stage6(2)(48 downto 47) <=partial_products(9).value(30 downto 29);
stage6(2)(50 downto 49) <=partial_products(10).value(30 downto 29);
stage6(2)(52 downto 51) <=partial_products(11).value(30 downto 29);
stage6(2)(54 downto 53) <=partial_products(12).value(30 downto 29);
stage6(2)(56 downto 55) <=partial_products(13).value(30 downto 29);
stage6(2)(58 downto 57) <=partial_products(14).value(30 downto 29);
stage6(2)(60 downto 59) <=partial_products(15).value(30 downto 29);
stage6(2)(62 downto 61) <=partial_products(16).value(30 downto 29);
--stage6 line3
stage6(3)(4)            <=partial_products(2).sign;
stage6(3)(35 downto 6)  <=partial_products(3).value(29 downto 0);
stage6(3)(36)           <=partial_products(4).value(28);
stage6(3)(38 downto 37) <=partial_products(5).value(28 downto 27);
stage6(3)(40 downto 39) <=partial_products(6).value(28 downto 27);
stage6(3)(42 downto 41) <=partial_products(7).value(28 downto 27);
stage6(3)(44 downto 43) <=partial_products(8).value(28 downto 27);
stage6(3)(46 downto 45) <=partial_products(9).value(28 downto 27);
stage6(3)(48 downto 47) <=partial_products(10).value(28 downto 27);
stage6(3)(50 downto 49) <=partial_products(11).value(28 downto 27);
stage6(3)(52 downto 51) <=partial_products(12).value(28 downto 27);
stage6(3)(54 downto 53) <=partial_products(13).value(28 downto 27);
stage6(3)(56 downto 55) <=partial_products(14).value(28 downto 27);
stage6(3)(58 downto 57) <=partial_products(15).value(28 downto 27);
stage6(3)(60 downto 59) <=partial_products(16).value(28 downto 27);
--stage6 line4
stage6(4)(6)            <=partial_products(3).sign;
stage6(4)(35 downto 8)  <=partial_products(4).value(27 downto 0);
stage6(4)(36)           <=partial_products(5).value(26);
stage6(4)(38 downto 37) <=partial_products(6).value(26 downto 25);
stage6(4)(40 downto 39) <=partial_products(7).value(26 downto 25);
stage6(4)(42 downto 41) <=partial_products(8).value(26 downto 25);
stage6(4)(44 downto 43) <=partial_products(9).value(26 downto 25);
stage6(4)(46 downto 45) <=partial_products(10).value(26 downto 25);
stage6(4)(48 downto 47) <=partial_products(11).value(26 downto 25);
stage6(4)(50 downto 49) <=partial_products(12).value(26 downto 25);
stage6(4)(52 downto 51) <=partial_products(13).value(26 downto 25);
stage6(4)(54 downto 53) <=partial_products(14).value(26 downto 25);
stage6(4)(56 downto 55) <=partial_products(15).value(26 downto 25);
stage6(4)(58 downto 57) <=partial_products(16).value(26 downto 25);
--stage6 line5
stage6(5)(8)            <=partial_products(4).sign;
stage6(5)(35 downto 10) <=partial_products(5).value(25 downto 0);
stage6(5)(36)           <=partial_products(6).value(24);
```

```
stage6(5)(38 downto 37) <=partial_products(7).value(24 downto 23);
stage6(5)(40 downto 39) <=partial_products(8).value(24 downto 23);
stage6(5)(42 downto 41) <=partial_products(9).value(24 downto 23);
stage6(5)(44 downto 43) <=partial_products(10).value(24 downto 23);
stage6(5)(46 downto 45) <=partial_products(11).value(24 downto 23);
stage6(5)(48 downto 47) <=partial_products(12).value(24 downto 23);
stage6(5)(50 downto 49) <=partial_products(13).value(24 downto 23);
stage6(5)(52 downto 51) <=partial_products(14).value(24 downto 23);
stage6(5)(54 downto 53) <=partial_products(15).value(24 downto 23);
stage6(5)(56 downto 55) <=partial_products(16).value(24 downto 23);
--stage6 line6
stage6(6)(10)           <=partial_products(5).sign;
stage6(6)(35 downto 12) <=partial_products(6).value(23 downto 0);
stage6(6)(36)           <=partial_products(7).value(22);
stage6(6)(38 downto 37) <=partial_products(8).value(22 downto 21);
stage6(6)(40 downto 39) <=partial_products(9).value(22 downto 21);
stage6(6)(42 downto 41) <=partial_products(10).value(22 downto 21);
stage6(6)(44 downto 43) <=partial_products(11).value(22 downto 21);
stage6(6)(46 downto 45) <=partial_products(12).value(22 downto 21);
stage6(6)(48 downto 47) <=partial_products(13).value(22 downto 21);
stage6(6)(50 downto 49) <=partial_products(14).value(22 downto 21);
stage6(6)(52 downto 51) <=partial_products(15).value(22 downto 21);
stage6(6)(54 downto 53) <=partial_products(16).value(22 downto 21);
--stage6 line7
stage6(7)(12)           <=partial_products(6).sign;
stage6(7)(35 downto 14) <=partial_products(7).value(21 downto 0);
stage6(7)(36)           <=partial_products(8).value(20);
stage6(7)(38 downto 37) <=partial_products(9).value(20 downto 19);
stage6(7)(40 downto 39) <=partial_products(10).value(20 downto 19);
stage6(7)(42 downto 41) <=partial_products(11).value(20 downto 19);
stage6(7)(44 downto 43) <=partial_products(12).value(20 downto 19);
stage6(7)(46 downto 45) <=partial_products(13).value(20 downto 19);
stage6(7)(48 downto 47) <=partial_products(14).value(20 downto 19);
stage6(7)(50 downto 49) <=partial_products(15).value(20 downto 19);
stage6(7)(52 downto 51) <=partial_products(16).value(20 downto 19);
--stage6 line8
stage6(8)(14)           <=partial_products(7).sign;
stage6(8)(35 downto 16) <=partial_products(8).value(19 downto 0);
stage6(8)(36)           <=partial_products(9).value(18);
stage6(8)(38 downto 37) <=partial_products(10).value(18 downto 17);
stage6(8)(40 downto 39) <=partial_products(11).value(18 downto 17);
stage6(8)(42 downto 41) <=partial_products(12).value(18 downto 17);
stage6(8)(44 downto 43) <=partial_products(13).value(18 downto 17);
stage6(8)(46 downto 45) <=partial_products(14).value(18 downto 17);
stage6(8)(48 downto 47) <=partial_products(15).value(18 downto 17);
stage6(8)(50 downto 49) <=partial_products(16).value(18 downto 17);
--stage6 line9
stage6(9)(16)           <=partial_products(8).sign;
stage6(9)(35 downto 18) <=partial_products(9).value(17 downto 0);
stage6(9)(36)           <=partial_products(10).value(16);
stage6(9)(38 downto 37) <=partial_products(11).value(16 downto 15);
stage6(9)(40 downto 39) <=partial_products(12).value(16 downto 15);
stage6(9)(42 downto 41) <=partial_products(13).value(16 downto 15);
stage6(9)(44 downto 43) <=partial_products(14).value(16 downto 15);
stage6(9)(46 downto 45) <=partial_products(15).value(16 downto 15);
stage6(9)(48 downto 47) <=partial_products(16).value(16 downto 15);
--stage6 line10
stage6(10)(18)           <=partial_products(9).sign;
stage6(10)(35 downto 20)<=partial_products(10).value(15 downto 0);
stage6(10)(36)           <=partial_products(11).value(14);
stage6(10)(38 downto 37)<=partial_products(12).value(14 downto 13);
stage6(10)(40 downto 39)<=partial_products(13).value(14 downto 13);
stage6(10)(42 downto 41)<=partial_products(14).value(14 downto 13);
stage6(10)(44 downto 43)<=partial_products(15).value(14 downto 13);
stage6(10)(46 downto 45)<=partial_products(16).value(14 downto 13);
--stage6 line11
stage6(11)(20)           <=partial_products(10).sign;
stage6(11)(35 downto 22)<=partial_products(11).value(13 downto 0);
stage6(11)(36)           <=partial_products(12).value(12);
stage6(11)(38 downto 37)<=partial_products(13).value(12 downto 11);
stage6(11)(40 downto 39)<=partial_products(14).value(12 downto 11);
stage6(11)(42 downto 41)<=partial_products(15).value(12 downto 11);
```

```
stage6(11)(44 downto 43)<=partial_products(16).value(12 downto 11);
--stage6 line12
stage6(12)(22)          <=partial_products(11).sign;
stage6(12)(35 downto 24)<=partial_products(12).value(11 downto 0);
stage6(12)(36)          <=partial_products(13).value(10);
stage6(12)(38 downto 37)<=partial_products(14).value(10 downto 9);
stage6(12)(40 downto 39)<=partial_products(15).value(10 downto 9);
stage6(12)(42 downto 41)<=partial_products(16).value(10 downto 9);
--stage6 line13
stage6(13)(24)          <=partial_products(12).sign;
stage6(13)(35 downto 26)<=partial_products(13).value(9 downto 0);
stage6(13)(36)          <=partial_products(14).value(8);
stage6(13)(38 downto 37)<=partial_products(15).value(8 downto 7);
stage6(13)(40 downto 39)<=partial_products(16).value(8 downto 7);
--stage6 line14
stage6(14)(26)          <=partial_products(13).sign;
stage6(14)(35 downto 28)<=partial_products(14).value(7 downto 0);
stage6(14)(36)          <=partial_products(15).value(6);
stage6(14)(38 downto 37)<=partial_products(16).value(6 downto 5);
--stage6 line15
stage6(15)(28)          <=partial_products(14).sign;
stage6(15)(35 downto 30)<=partial_products(15).value(5 downto 0);
stage6(15)(36)          <=partial_products(16).value(4);
--stage6 line16
stage6(16)(30)          <=partial_products(15).sign;
stage6(16)(35 downto 32)<=partial_products(16).value(3 downto 0);

---------------------------------------------FROM STAGE6 TO STAGE5
     ↪ -------------------------------------------------------

--FAs

FA_6to5_generate0: for i in 26 to 41 generate
    stage5(0)(i)<=FA_sum(stage6(0)(i), stage6(1)(i), stage6(2)(i));
    stage5(1)(i+1)<=FA_cout(stage6(0)(i), stage6(1)(i), stage6(2)(i));
end generate FA_6to5_generate0;

FA_6to5_generate1: for i in 28 to 39 generate
    stage5(2)(i)<=FA_sum(stage6(3)(i), stage6(4)(i), stage6(5)(i));
    stage5(3)(i+1)<=FA_cout(stage6(3)(i), stage6(4)(i), stage6(5)(i));
end generate FA_6to5_generate1;

FA_6to5_generate2: for i in 30 to 37 generate
    stage5(4)(i)<=FA_sum(stage6(6)(i), stage6(7)(i), stage6(8)(i));
    stage5(5)(i+1)<=FA_cout(stage6(6)(i), stage6(7)(i), stage6(8)(i));
end generate FA_6to5_generate2;

FA_6to5_generate3: for i in 32 to 35 generate
    stage5(6)(i)<=FA_sum(stage6(9)(i), stage6(10)(i), stage6(11)(i));
    stage5(7)(i+1)<=FA_cout(stage6(9)(i), stage6(10)(i), stage6(11)(i));
end generate FA_6to5_generate3;

--HAs

HA_6to5_generate0: for i in 24 to 25 generate
    stage5(0)(i)<=HA_sum(stage6(0)(i), stage6(1)(i));
    stage5(1)(i+1)<=HA_cout(stage6(0)(i), stage6(1)(i));
end generate HA_6to5_generate0;

HA_6to5_generate1: for i in 26 to 27 generate
    stage5(2)(i)<=HA_sum(stage6(3)(i), stage6(4)(i));
    stage5(3)(i+1)<=HA_cout(stage6(3)(i), stage6(4)(i));
end generate HA_6to5_generate1;

HA_6to5_generate2: for i in 28 to 29 generate
    stage5(4)(i)<=HA_sum(stage6(6)(i), stage6(7)(i));
    stage5(5)(i+1)<=HA_cout(stage6(6)(i), stage6(7)(i));
end generate HA_6to5_generate2;

HA_6to5_generate3: for i in 30 to 31 generate
    stage5(6)(i)<=HA_sum(stage6(9)(i), stage6(10)(i));
    stage5(7)(i+1)<=HA_cout(stage6(9)(i), stage6(10)(i));
end generate HA_6to5_generate3;

stage5(6)(36)<=HA_sum(stage6(9)(36), stage6(10)(36));
stage5(6)(37)<=HA_cout(stage6(9)(36), stage6(10)(36));

stage5(4)(38)<=HA_sum(stage6(6)(38), stage6(7)(38));
stage5(4)(39)<=HA_cout(stage6(6)(38), stage6(7)(38));

stage5(2)(40)<=HA_sum(stage6(3)(40), stage6(4)(40));
stage5(2)(41)<=HA_cout(stage6(3)(40), stage6(4)(40));

stage5(0)(42)<=HA_sum(stage6(0)(42), stage6(1)(42));
stage5(0)(43)<=HA_cout(stage6(0)(42), stage6(1)(42));

--direct connections

connection_6to5_column0to23_generate_ext: for j in 0 to 23 generate
    connection_6to5_column0to23_generate_in: for i in 0 to 12 generate
        stage5(i)(j)<=stage6(i)(j);
    end generate connection_6to5_column0to23_generate_in;
end generate connection_6to5_column0to23_generate_ext;

connection_6to5_column24_generate: for i in 1 to 12 generate
    stage5(i)(24)<=stage6(i+1)(24);
end generate connection_6to5_column24_generate;

connection_6to5_column25_generate: for i in 2 to 12 generate
    stage5(i)(25)<=stage6(i)(25);
end generate connection_6to5_column25_generate;

connection_6to5_column26_generate: for i in 3 to 12 generate
    stage5(i)(26)<=stage6(i+2)(26);
end generate connection_6to5_column26_generate;

connection_6to5_column27_generate: for i in 4 to 12 generate
    stage5(i)(27)<=stage6(i+1)(27);
end generate connection_6to5_column27_generate;

connection_6to5_column28_generate: for i in 5 to 12 generate
    stage5(i)(28)<=stage6(i+3)(28);
end generate connection_6to5_column28_generate;

connection_6to5_column29_generate: for i in 6 to 12 generate
    stage5(i)(29)<=stage6(i+2)(29);
end generate connection_6to5_column29_generate;

connection_6to5_column30_generate: for i in 7 to 12 generate
    stage5(i)(30)<=stage6(i+4)(30);
end generate connection_6to5_column30_generate;

connection_6to5_column31_generate: for i in 8 to 12 generate --12
    stage5(i)(31)<=stage6(i+3)(31);
end generate connection_6to5_column31_generate;

connection_6to5_column32to35_generate_ext: for j in 32 to 35 generate
    connection_6to5_column32to35_generate_in: for i in 8 to 12 generate
        stage5(i)(j)<=stage6(i+4)(j);
    end generate connection_6to5_column32to35_generate_in;
end generate connection_6to5_column32to35_generate_ext;

connection_6to5_column36_generate: for i in 8 to 12 generate
    stage5(i)(36)<=stage6(i+3)(36);
end generate connection_6to5_column36_generate;

connection_6to5_column37_generate: for i in 7 to 12 generate
    stage5(i)(37)<=stage6(i+2)(37);
end generate connection_6to5_column37_generate;

connection_6to5_column38_generate: for i in 6 to 12 generate
    stage5(i)(38)<=stage6(i+2)(38);
end generate connection_6to5_column38_generate;
```

```
connection_6to5_column39_generate: for i in 5 to 12 generate
    stage5(i)(39)<=stage6(i+1)(39);
end generate connection_6to5_column39_generate;

connection_6to5_column40_generate: for i in 4 to 12 generate
    stage5(i)(40)<=stage6(i+1)(40);
end generate connection_6to5_column40_generate;

connection_6to5_column41_generate: for i in 3 to 12 generate
    stage5(i)(41)<=stage6(i)(41);
end generate connection_6to5_column41_generate;

connection_6to5_column42_generate: for i in 2 to 12 generate
    stage5(i)(42)<=stage6(i)(42);
end generate connection_6to5_column42_generate;

connection_6to5_column43_generate: for i in 1 to 12 generate
    stage5(i)(43)<=stage6(i-1)(43);
end generate connection_6to5_column43_generate;

connection_6to5_column44to63_generate_ext: for j in 44 to 63 generate
    connection_6to5_column44to63_generate_in: for i in 0 to 12 generate
        stage5(i)(j)<=stage6(i)(j);
    end generate connection_6to5_column44to63_generate_in;
end generate connection_6to5_column44to63_generate_ext;

-----------------------------------------------------FROM STAGE5 to STAGE4
    ↪ -----------------------------------------------------------------------

--FAs

FA_5to4_generate0: for i in 18 to 49 generate
    stage4(0)(i)<=FA_sum(stage5(0)(i), stage5(1)(i), stage5(2)(i));
    stage4(1)(i+1)<=FA_cout(stage5(0)(i), stage5(1)(i), stage5(2)(i));
end generate FA_5to4_generate0;

FA_5to4_generate1: for i in 20 to 47 generate
    stage4(2)(i)<=FA_sum(stage5(3)(i), stage5(4)(i), stage5(5)(i));
    stage4(3)(i+1)<=FA_cout(stage5(3)(i), stage5(4)(i), stage5(5)(i));
end generate FA_5to4_generate1;

FA_5to4_generate2: for i in 22 to 45 generate
    stage4(4)(i)<=FA_sum(stage5(6)(i), stage5(7)(i), stage5(8)(i));
    stage4(5)(i+1)<=FA_cout(stage5(6)(i), stage5(7)(i), stage5(8)(i));
end generate FA_5to4_generate2;

FA_5to4_generate3: for i in 24 to 43 generate
    stage4(6)(i)<=FA_sum(stage5(9)(i), stage5(10)(i), stage5(11)(i));
    stage4(7)(i+1)<=FA_cout(stage5(9)(i), stage5(10)(i), stage5(11)(i));
end generate FA_5to4_generate3;

--HAs

HA_5to4_generate0: for i in 16 to 17 generate
    stage4(0)(i)<=HA_sum(stage5(0)(i), stage5(1)(i));
    stage4(1)(i+1)<=HA_cout(stage5(0)(i), stage5(1)(i));
end generate HA_5to4_generate0;

HA_5to4_generate1: for i in 18 to 19 generate
    stage4(2)(i)<=HA_sum(stage5(3)(i), stage5(4)(i));
    stage4(3)(i+1)<=HA_cout(stage5(3)(i), stage5(4)(i));
end generate HA_5to4_generate1;

HA_5to4_generate2: for i in 20 to 21 generate
    stage4(4)(i)<=HA_sum(stage5(6)(i), stage5(7)(i));
    stage4(5)(i+1)<=HA_cout(stage5(6)(i), stage5(7)(i));
end generate HA_5to4_generate2;

HA_5to4_generate3: for i in 22 to 23 generate
    stage4(6)(i)<=HA_sum(stage5(9)(i), stage5(10)(i));
    stage4(7)(i+1)<=HA_cout(stage5(9)(i), stage5(10)(i));
end generate HA_5to4_generate3;
```

```
stage4(6)(44)<=HA_sum(stage5(9)(44), stage5(10)(44));
stage4(6)(45)<=HA_cout(stage5(9)(44), stage5(10)(44));

stage4(4)(46)<=HA_sum(stage5(6)(46), stage5(7)(46));
stage4(4)(47)<=HA_cout(stage5(6)(46), stage5(7)(46));

stage4(2)(48)<=HA_sum(stage5(3)(48), stage5(4)(48));
stage4(2)(49)<=HA_cout(stage5(3)(48), stage5(4)(48));

stage4(0)(50)<=HA_sum(stage5(0)(50), stage5(1)(50));
stage4(0)(51)<=HA_cout(stage5(0)(50), stage5(1)(50));

--direct connections

connection_5to4_column0to15_generate_ext: for j in 0 to 15 generate
    connection_5to4_column0to15_generate_in: for i in 0 to 8 generate
        stage4(i)(j)<=stage5(i)(j);
    end generate connection_5to4_column0to15_generate_in;
end generate connection_5to4_column0to15_generate_ext;

connection_5to4_column16_generate: for i in 1 to 8 generate
    stage4(i)(16)<=stage5(i+1)(16);
end generate connection_5to4_column16_generate;

connection_5to4_column17_generate: for i in 2 to 8 generate
    stage4(i)(17)<=stage5(i)(17);
end generate connection_5to4_column17_generate;

connection_5to4_column18_generate: for i in 3 to 8 generate
    stage4(i)(18)<=stage5(i+2)(18);
end generate connection_5to4_column18_generate;

connection_5to4_column19_generate: for i in 4 to 8 generate
    stage4(i)(19)<=stage5(i+1)(19);
end generate connection_5to4_column19_generate;

connection_5to4_column20_generate: for i in 5 to 8 generate
    stage4(i)(20)<=stage5(i+3)(20);
end generate connection_5to4_column20_generate;

connection_5to4_column21_generate: for i in 6 to 8 generate
    stage4(i)(21)<=stage5(i+2)(21);
end generate connection_5to4_column21_generate;

connection_5to4_column22_generate: for i in 7 to 8 generate
    stage4(i)(22)<=stage5(i+4)(22);
end generate connection_5to4_column22_generate;

stage4(8)(23)<=stage5(11)(23);

connection_5to4_column24to43_generate: for j in 24 to 43 generate
    stage4(8)(j)<=stage5(12)(j);
end generate connection_5to4_column24to43_generate;

-- connection_5to4_column24to30_generate: for j in 24 to 30 generate
    -- stage4(8)(j)<=stage5(12)(j);
-- end generate connection_5to4_column24to30_generate;
-- stage4(8)(31)<=stage6(15)(31);
-- connection_5to4_column32to43_generate: for j in 32 to 43 generate
    -- stage4(8)(j)<=stage5(12)(j);
-- end generate connection_5to4_column32to43_generate;

stage4(8)(44)<=stage5(11)(44);

connection_5to4_column45_generate: for i in 7 to 8 generate
    stage4(i)(45)<=stage5(i+2)(45);
end generate connection_5to4_column45_generate;

connection_5to4_column46_generate: for i in 6 to 8 generate
    stage4(i)(46)<=stage5(i+2)(46);
end generate connection_5to4_column46_generate;
```

```
connection_5to4_column47_generate: for i in 5 to 8 generate
    stage4(i)(47)<=stage5(i+1)(47);
end generate connection_5to4_column47_generate;

connection_5to4_column48_generate: for i in 4 to 8 generate
    stage4(i)(48)<=stage5(i+1)(48);
end generate connection_5to4_column48_generate;

connection_5to4_column49_generate: for i in 3 to 8 generate
    stage4(i)(49)<=stage5(i)(49);
end generate connection_5to4_column49_generate;

connection_5to4_column50_generate: for i in 2 to 8 generate
    stage4(i)(50)<=stage5(i)(50);
end generate connection_5to4_column50_generate;

connection_5to4_column51_generate: for i in 1 to 8 generate
    stage4(i)(51)<=stage5(i-1)(51);
end generate connection_5to4_column51_generate;

connection_5to4_column52to63_generate_ext: for j in 52 to 63 generate
    connection_5to4_column52to63_generate_in: for i in 0 to 8 generate
        stage4(i)(j)<=stage5(i)(j);
    end generate connection_5to4_column52to63_generate_in;
end generate connection_5to4_column52to63_generate_ext;

-------------------------------------------------------FROM STAGE4 TO STAGE3
    ↪ ---------------------------------------------------------------------------

--FAs

FA_4to3_generate0: for i in 12 to 55 generate
    stage3(0)(i)<=FA_sum(stage4(0)(i), stage4(1)(i), stage4(2)(i));
    stage3(1)(i+1)<=FA_cout(stage4(0)(i), stage4(1)(i), stage4(2)(i));
end generate FA_4to3_generate0;

FA_4to3_generate1: for i in 14 to 53 generate
    stage3(2)(i)<=FA_sum(stage4(3)(i), stage4(4)(i), stage4(5)(i));
    stage3(3)(i+1)<=FA_cout(stage4(3)(i), stage4(4)(i), stage4(5)(i));
end generate FA_4to3_generate1;

FA_4to3_generate2: for i in 16 to 51 generate
    stage3(4)(i)<=FA_sum(stage4(6)(i), stage4(7)(i), stage4(8)(i));
    stage3(5)(i+1)<=FA_cout(stage4(6)(i), stage4(7)(i), stage4(8)(i));
end generate FA_4to3_generate2;

--HAs

HA_4to3_generate0: for i in 10 to 11 generate
    stage3(0)(i)<=HA_sum(stage4(0)(i), stage4(1)(i));
    stage3(1)(i+1)<=HA_cout(stage4(0)(i), stage4(1)(i));
end generate HA_4to3_generate0;

HA_4to3_generate1: for i in 12 to 13 generate
    stage3(2)(i)<=HA_sum(stage4(3)(i), stage4(4)(i));
    stage3(3)(i+1)<=HA_cout(stage4(3)(i), stage4(4)(i));
end generate HA_4to3_generate1;

HA_4to3_generate2: for i in 14 to 15 generate
    stage3(4)(i)<=HA_sum(stage4(6)(i), stage4(7)(i));
    stage3(5)(i+1)<=HA_cout(stage4(6)(i), stage4(7)(i));
end generate HA_4to3_generate2;

stage3(4)(52)<=HA_sum(stage4(6)(52), stage4(7)(52));
stage3(4)(53)<=HA_cout(stage4(6)(52), stage4(7)(52));

stage3(2)(54)<=HA_sum(stage4(3)(54), stage4(4)(54));
stage3(2)(55)<=HA_cout(stage4(3)(54), stage4(4)(54));

stage3(0)(56)<=HA_sum(stage4(0)(56), stage4(1)(56));
stage3(0)(57)<=HA_cout(stage4(0)(56), stage4(1)(56));
```

```
--direct connections

connection_4to3_column0to9_generate_ext: for j in 0 to 9 generate
    connection_4to3_column0to9_generate_in: for i in 0 to 5 generate
        stage3(i)(j)<=stage4(i)(j);
    end generate connection_4to3_column0to9_generate_in;
end generate connection_4to3_column0to9_generate_ext;

connection_4to3_column10_generate: for i in 1 to 5 generate
    stage3(i)(10)<=stage4(i+1)(10);
end generate connection_4to3_column10_generate;

connection_4to3_column11_generate: for i in 2 to 5 generate
    stage3(i)(11)<=stage4(i)(11);
end generate connection_4to3_column11_generate;

connection_4to3_column12_generate: for i in 3 to 5 generate
    stage3(i)(12)<=stage4(i+2)(12);
end generate connection_4to3_column12_generate;

connection_4to3_column13_generate: for i in 4 to 5 generate
    stage3(i)(13)<=stage4(i+1)(13);
end generate connection_4to3_column13_generate;

stage3(5)(14)<=stage4(8)(14);

stage3(5)(53)<=stage4(6)(53);

connection_4to3_column54_generate: for i in 4 to 5 generate
    stage3(i)(54)<=stage4(i+1)(54);
end generate connection_4to3_column54_generate;

connection_4to3_column55_generate: for i in 3 to 5 generate
    stage3(i)(55)<=stage4(i)(55);
end generate connection_4to3_column55_generate;

connection_4to3_column56_generate: for i in 2 to 5 generate
    stage3(i)(56)<=stage4(i)(56);
end generate connection_4to3_column56_generate;

connection_4to3_column57_generate: for i in 1 to 5 generate
    stage3(i)(57)<=stage4(i-1)(57);
end generate connection_4to3_column57_generate;

connection_4to3_column58to63_generate_ext: for j in 58 to 63 generate
    connection_4to3_column58to63_generate_in: for i in 0 to 5 generate
        stage3(i)(j)<=stage4(i)(j);
    end generate connection_4to3_column58to63_generate_in;
end generate connection_4to3_column58to63_generate_ext;

---------------------------------------------------------FROM STAGE3 TO STAGE2
    ↪ --------------------------------------------------------------------------------

--FAs

FA_3to2_generate0: for i in 8 to 59 generate
    stage2(0)(i)<=FA_sum(stage3(0)(i), stage3(1)(i), stage3(2)(i));
    stage2(1)(i+1)<=FA_cout(stage3(0)(i), stage3(1)(i), stage3(2)(i));
end generate FA_3to2_generate0;

FA_3to2_generate1: for i in 10 to 57 generate
    stage2(2)(i)<=FA_sum(stage3(3)(i), stage3(4)(i), stage3(5)(i));
    stage2(3)(i+1)<=FA_cout(stage3(3)(i), stage3(4)(i), stage3(5)(i));
end generate FA_3to2_generate1;

--HAs

HA_3to2_generate0: for i in 6 to 7 generate
    stage2(0)(i)<=HA_sum(stage3(0)(i), stage3(1)(i));
    stage2(1)(i+1)<=HA_cout(stage3(0)(i), stage3(1)(i));
end generate HA_3to2_generate0;
```

```
HA_3to2_generate1: for i in 8 to 9 generate
    stage2(2)(i)<=HA_sum(stage3(3)(i), stage3(4)(i));
    stage2(3)(i+1)<=HA_cout(stage3(3)(i), stage3(4)(i));
end generate HA_3to2_generate1;

stage2(2)(58)<=HA_sum(stage3(3)(58), stage3(4)(58));
stage2(2)(59)<=HA_cout(stage3(3)(58), stage3(4)(58));

stage2(0)(60)<=HA_sum(stage3(0)(60), stage3(1)(60));
stage2(0)(61)<=HA_cout(stage3(0)(60), stage3(1)(60));

--direct connections

connection_3to2_column0to5_generate_ext: for j in 0 to 5 generate
    connection_3to2_column0to5_generate_in: for i in 0 to 3 generate
        stage2(i)(j)<=stage3(i)(j);
    end generate connection_3to2_column0to5_generate_in;
end generate connection_3to2_column0to5_generate_ext;

connection_3to2_column6_generate: for i in 1 to 3 generate
    stage2(i)(6)<=stage3(i+1)(6);
end generate connection_3to2_column6_generate;

connection_3to2_column7_generate: for i in 2 to 3 generate
    stage2(i)(7)<=stage3(i)(7);
end generate connection_3to2_column7_generate;

stage2(3)(8)<=stage3(5)(8);

stage2(3)(59)<=stage3(3)(59);

connection_3to2_column60_generate: for i in 2 to 3 generate
    stage2(i)(60)<=stage3(i)(60);
end generate connection_3to2_column60_generate;

connection_3to2_column61_generate: for i in 1 to 3 generate
    stage2(i)(61)<=stage3(i-1)(61);
end generate connection_3to2_column61_generate;

connection_3to2_column62to63_generate_ext: for j in 62 to 63 generate
    connection_3to2_column62to63_generate_in: for i in 0 to 3 generate
        stage2(i)(j)<=stage3(i)(j);
    end generate connection_3to2_column62to63_generate_in;
end generate connection_3to2_column62to63_generate_ext;

--------------------------------------------------------FROM STAGE2 TO STAGE1
    ↪ ------------------------------------------------------------------------

--FAs

FA_2to1_generate: for i in 6 to 61 generate
    stage1(0)(i)<=FA_sum(stage2(0)(i), stage2(1)(i), stage2(2)(i));
    stage1(1)(i+1)<=FA_cout(stage2(0)(i), stage2(1)(i), stage2(2)(i));
end generate FA_2to1_generate;

--HAs

HA_2to1_generate: for i in 4 to 5 generate
    stage1(0)(i)<=HA_sum(stage2(0)(i), stage2(1)(i));
    stage1(1)(i+1)<=HA_cout(stage2(0)(i), stage2(1)(i));
end generate HA_2to1_generate;

stage1(0)(62)<=HA_sum(stage2(0)(62), stage2(1)(62));
stage1(0)(63)<=HA_cout(stage2(0)(62), stage2(1)(62));

--direct connections

connection_2to1_column0to3_generate_ext: for j in 0 to 3 generate
    connection_2to1_column0to3_generate_in: for i in 0 to 2 generate
        stage1(i)(j)<=stage2(i)(j);
    end generate connection_2to1_column0to3_generate_in;
    end generate connection_2to1_column0to3_generate_ext;

    connection_2to1_column4_generate: for i in 1 to 2 generate
        stage1(i)(4)<=stage2(i+1)(4);
    end generate connection_2to1_column4_generate;

    stage1(2)(5)<=stage2(2)(5);

    connection_2to1_column6to61_generate_ext: for j in 6 to 61 generate
            stage1(2)(j)<=stage2(3)(j);
    end generate connection_2to1_column6to61_generate_ext;

    stage1(2)(62)<=stage2(2)(62);

    connection_2to1_column63_generate: for i in 1 to 2 generate
        stage1(i)(63)<=stage2(i-1)(63);
    end generate connection_2to1_column63_generate;

    ------------------------------------------------------------FROM STAGE1 to STAGE0
        ↪ ---------------------------------------------------------------

    --FAs

    FA_1to0_generate: for i in 4 to 63 generate
        stage0(0)(i)<=FA_sum(stage1(0)(i), stage1(1)(i), stage1(2)(i));
        stage0(1)(i+1)<=FA_cout(stage1(0)(i), stage1(1)(i), stage1(2)(i));
    end generate FA_1to0_generate;

    --HAs

    HA_1to0_generate: for i in 2 to 3 generate
        stage0(0)(i)<=HA_sum(stage1(0)(i), stage1(1)(i));
        stage0(1)(i+1)<=HA_cout(stage1(0)(i), stage1(1)(i));
    end generate HA_1to0_generate;

    --direct connections

    connection_1to0_column0_generate: for i in 0 to 1 generate
        stage0(i)(0)<=stage1(i)(0);
    end generate connection_1to0_column0_generate;

    stage0(0)(1)<=stage1(0)(1);
    stage0(1)(1)<='0';

    stage0(1)(2)<=stage1(2)(2);

    stage0(0)(64)<=stage0(0)(63);   --sign extension

    --------------------------------------------------------FROM STAGE0 TO FINAL SUM
        ↪ --------------------------------------------------------------------------------

    x_tmp <= std_logic_vector(unsigned(stage0(0))+unsigned(stage0(1)));
    x <= x_tmp(2*N-1 downto 0);
end architecture behavioral;
```

Listing A.5: pack_mult32_MBE_dadda.vhd

```
library ieee;
use ieee.std_logic_1164.all;

package pack_mult32_MBE_dadda is

    constant N: integer := 32;

    type std_logic_vector_intrange is array (integer range <>) of std_logic;    --standard logic
        ↪ vector with either positive or negative integer range

    type partial_product_type is record
        value   : std_logic_vector(N downto 0);     --value of the partial product in ones'
```

```
            ↪ complement
        sign    : std_logic;                        --0=pos 1=neg
    end record partial_product_type;

    type partial_products_type is array (integer range <>) of partial_product_type;

    function get_MBE_partial_product(
        multiplicand: std_logic_vector(N-1 downto 0);
        multiplier: std_logic_vector_intrange(N+1 downto -1);
        i: integer)
    return partial_product_type;

    type stage_type is array (integer range <>) of std_logic_vector(2*N downto 0);

    function HA_sum(
        A: std_logic;
        B: std_logic)
    return std_logic;

    function HA_cout(
        A: std_logic;
        B: std_logic)
    return std_logic;

    function FA_sum(
        A: std_logic;
        B: std_logic;
        C_in: std_logic)
    return std_logic;

    function FA_cout(
        A: std_logic;
        B: std_logic;
        C_in: std_logic)
    return std_logic;

end package pack_mult32_MBE_dadda;

package body pack_mult32_MBE_dadda is

    function get_MBE_partial_product (
        multiplicand: std_logic_vector(N-1 downto 0);
        multiplier: std_logic_vector_intrange(N+1 downto -1);
        i: integer)
        return partial_product_type is
        variable temp_partial_product: partial_product_type;
        constant zeros: std_logic_vector(N downto 0):=(others=>'0');
        constant ones: std_logic_vector(N downto 0):=(others=>'1');
        variable multiplicand_by_one: std_logic_vector(N downto 0);
        variable multiplicand_by_two: std_logic_vector(N downto 0);
        variable multiplicand_by_one_neg: std_logic_vector(N downto 0);
        variable multiplicand_by_two_neg: std_logic_vector(N downto 0);
    begin
        multiplicand_by_one(N-1 downto 0):=multiplicand;
        multiplicand_by_one(N):='0';
        multiplicand_by_two(0):='0';
        multiplicand_by_two(N downto 1):=multiplicand;
        multiplicand_by_one_neg:= not multiplicand_by_one;
        multiplicand_by_two_neg:= not multiplicand_by_two;
        if multiplier(i+1 downto i-1)="000" then
            temp_partial_product.value:=zeros;
            temp_partial_product.sign:='0';
        elsif multiplier(i+1 downto i-1)="001" or multiplier(i+1 downto i-1)="010" then
            temp_partial_product.value:=multiplicand_by_one;
            temp_partial_product.sign:='0';
        elsif multiplier(i+1 downto i-1)="011" then
            temp_partial_product.value:=multiplicand_by_two;
            temp_partial_product.sign:='0';
        elsif multiplier(i+1 downto i-1)="100" then
            temp_partial_product.value:=multiplicand_by_two_neg;
            temp_partial_product.sign:='1';
        elsif multiplier(i+1 downto i-1)="101" or multiplier(i+1 downto i-1)="110" then
```

```
            temp_partial_product.value:=multiplicand_by_one_neg;
            temp_partial_product.sign:='1';
        elsif multiplier(i+1 downto i-1)="111" then
            temp_partial_product.value:=ones;
            temp_partial_product.sign:='1';
        end if;
        return temp_partial_product;
    end function get_MBE_partial_product;

    function HA_sum(
        A: std_logic;
        B: std_logic)
        return std_logic is
    begin
        return A xor B;
    end function HA_sum;

    function HA_cout(
        A: std_logic;
        B: std_logic)
        return std_logic is
    begin
        return A and B;
    end function HA_cout;

    function FA_sum(
        A: std_logic;
        B: std_logic;
        C_in: std_logic)
        return std_logic is
    begin
        return A xor B xor C_in;
    end function FA_sum;

    function FA_cout(
        A: std_logic;
        B: std_logic;
        C_in: std_logic)
        return std_logic is
    begin
        return (A and B) or (A and C_in) or (B and C_in);
    end function FA_cout;

end package body pack_mult32_MBE_dadda;
```

Listing A.6: reg.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;

-- register description
entity reg is
    Generic (N: positive:= 1 );                              --  number of bits
    Port(   D       : In    std_logic_vector(N-1 downto 0);      --  data input
            Q       : Out   std_logic_vector(N-1 downto 0);      --  data output
            CLK     : In    std_logic                            -- clock signal
    );
end entity reg;

architecture behavioral of reg is

begin
    PSYNCH: process(CLK)
    begin
        if rising_edge(CLK) then
                Q <= D;             -- writes the input on the output
        end if;
    end process;
end behavioral;
```

APPENDIX B

Testbench listings

Listing B.1: tb_FPmult.v

```
//‘timescale 1 ns
// testbench for the multplier
module tb ();

    wire [31:0] op1;
    wire clk;
    wire [31:0] res;
    wire w_enable;

    FPmul DUT(
        .FP_A(op1),
        .FP_B(op1),
        .clk(clk) ,
        .FP_Z(res)
        );

    data_maker data_maker_inst(
        .CLK(clk),
        .DATA(op1),
        .ENDSIM(w_enable)
    );

    data_sink data_sink_inst(
        .CLK(clk),
        .VIN(w_enable),
        .DIN(res)
    );

    clk_gen clk_gen_inst(
        .CLK(clk)
    );

endmodule
```

Listing B.2: clk_gen.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

-- used to generate the clock
entity clk_gen is
  port (
    CLK     : out std_logic);
end clk_gen;

architecture beh of clk_gen is

    constant Ts : time := 5 ns;
    signal CLK_i : std_logic;

begin
    process
    begin  -- process
        if (CLK_i = 'U') then
            CLK_i <= '0';
        else
            CLK_i <= not(CLK_i);
        end if;
        wait for Ts/2;
    end process;
    CLK <= CLK_i;
end beh;
```

Listing B.3: data_maker.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_textio.all;

library std;
use std.textio.all;
-- Used to read from a file the values and pass them to the DUT

entity data_maker is
  port (
    CLK     : in  std_logic;
    ENDSIM  : out std_logic;
    DATA    : out std_logic_vector(31 downto 0));
end data_maker;

architecture beh of data_maker is

    constant stage_pipeline : integer := 6;         -- change this value to write correctly the
         ↪ output file

    signal sENDSIM : std_logic;                     -- goes to 0 when the input file is
         ↪ completely read
    signal vin : std_logic;
    signal stop_write : std_logic;

BEGIN

    process (CLK)
    file fp : text open read_mode is "./fp_samples.hex";        --input file
    variable ptr : line;
    variable val : std_logic_vector(31 downto 0);
    begin
        if CLK'event and CLK = '1' then
            if (not(endfile(fp))) then
                readline(fp, ptr);                              -- read one value
                hread(ptr, val);
                sENDSIM <= '0';
            else
                sENDSIM <= '1';
            end if;
        DATA <= val;
        end if;
    end process;
    validation_write: process(clk)
    variable empty_pipe : integer:= 0;
    variable cc : integer:=0;
    begin
        if rising_edge(clk) then
        cc := cc+1;
            if cc <= stage_pipeline then            -- block the writing of first value until the
                ↪ pipeline is not full
                vin <= '0';
            else
                vin<='1';
            end if;

            if sENDSIM = '1' then                   -- stop the writing when all the operands are
                ↪ passed through the pipeline
            empty_pipe := empty_pipe+1;
            end if;

            if empty_pipe >= stage_pipeline then
                stop_write <= '0';
            else
                stop_write <='1';
            end if;
        end if;
    end process validation_write;

    ENDSIM <= vin and stop_write;

end beh;
```

Listing B.4: data_sink.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_textio.all;

library std;
use std.textio.all;

-- component used to write the results. It write only when the validation signal is equal to 1

entity data_sink is
  port (
    CLK     : in std_logic;                         -- clock signal
    VIN     : in std_logic;                         -- validation signal
    DIN     : in std_logic_vector(31 downto 0));    -- input data
end data_sink;

architecture beh of data_sink is

begin

    -- write the result in the file result; if it is not present in the folder of the project, it
        ↪ creates one and write it
    process (CLK)
        file res_fp : text open WRITE_MODE is "./results.txt";
        variable line_out : line;
    begin
        if rising_edge(CLK) then        -- rising clock edge
            if (VIN = '1') then
                write(line_out, DIN);
                writeline(res_fp, line_out);
            end if;
        end if;
    end process;

end beh;
```

Listing B.5: fp_prod.hex

```
00000000
3dc3910d
0bc80005
3f278ddf
0b100005
3f800000
0c440004
3f278ddf
0da20000
3dc3910d
```

Listing B.6: results.txt

```
00000000000000000000000000000000
00111101110000111001000100001101
00001011110010000000000000000101
00111111001001111000110111011111
00001011000100000000000000000101
00111111100000000000000000000000
00001100010001000000000000000100
00111111001001111000110111011111
00001101101000100000000000000000
00111101110000111001000100001101
```

Listing B.7: tb_mult32_MBE_dadda.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use work.pack_mult32_MBE_dadda.all;

entity tb_mult32_MBE_dadda is
end entity tb_mult32_MBE_dadda;

architecture test of tb_mult32_MBE_dadda is

    component mult32_MBE_dadda is
        port(
            a, b: in std_logic_vector(N-1 downto 0);
            x: out std_logic_vector(2*N-1 downto 0)
        );
    end component mult32_MBE_dadda;

    signal a,b: std_logic_vector(N-1 downto 0);
    signal x: std_logic_vector(2*N-1 downto 0);

begin

    DUT: mult32_MBE_dadda port map(a,b,x);

    stim_proc: process
    begin
        a<="00000000000000000000000000000001";
        b<="00000000000000000000000000000000";
        wait for 10 ns;
        b<="00000000000000000000000000000001";
        wait for 10 ns;
        a<="11111111111111111111111111111111";
        b<="00000000000000000000000000000000";
        wait for 10 ns;
        a<="00000000000000000000000000000001";
        b<="11111111111111111111111111111111";
        wait for 10 ns;
        a<="11111111111111111111111111111111";
        b<="11111111111111111111111111111111";
        wait for 10 ns;
        a<="10101010101010101010101010101010";
        b<="10101010101010101010101010101010";
        wait for 10 ns;
        a<="10101010101010101010101010101010";
        b<="01010101010101010101010101010101";
        wait for 10 ns;
        a<="10000000000000000000000000000000";
        b<="10000000000000000000000000000000";
        wait for 10 ns;
        a<="00000000000000000000000000000001";
        b<="00000000000000000000000000000000";

        wait;
    end process stim_proc;

end architecture test;
```

APPENDIX C

Synthesis script listings

Listing C.1: script_no_impl.rpt

```
#******************************************************************************
# Script used to synthesis the floating point multiplier
#******************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage2_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_pipeline_reg.vhd}
#elaborate top entity
elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) **********************
# create a clock signal to constraint the sequential paths
set WCP 1.6
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
#compile
compile
#timing and area report
report_timing > timing_no_impl.rpt
report_area > area_no_impl.rpt
```

Listing C.2: script_CSA.rpt

```
#******************************************************************************
# Script used to synthesis the floating point multiplier
#******************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage2_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_pipeline_reg.vhd}
#elaborate top entity
elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) ***********************
# create a clock signal to constraint the sequential paths
set WCP 4.1
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
set_implementation DW02_mult/csa [find cell *mult*]
#compile
compile
#timing and area report
report_timing > timing_CSA.rpt
report_area > area_CSA.rpt
```

Listing C.3: script_PPARCH.rpt

```
#*****************************************************************************
# Script used to synthesis the floating point multiplier
#*****************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage2_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_pipeline_reg.vhd}
#elaborate top entity
elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) ***********************
# create a clock signal to constraint the sequential paths
set WCP 1.60
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
set_implementation DW02_mult/pparch [find cell *mult*]
#compile
compile
#timing and area report
report_timing > timing_PPARCH.rpt
report_area > area_PPARCH.rpt
```

Listing C.4: script_additional_reg.tcl

```
#*****************************************************************************
# Script used to synthesis the floating point multiplier
#*****************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
```

```
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage2_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_pipeline_reg_stage2.vhd}
#elaborate top entity

elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) ***********************
# create a clock signal to constraint the sequential paths
set WCP 0.7
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
#compile
compile
#timing and area report
report_timing > timing_additional_reg.rpt
report_area > area_additional_reg.rpt
# optimize_register command
optimize_register
#timing and area report
report_timing > timing_additional_reg_retiming.rpt
report_area > area_additional_reg_retiming.rpt
```

Listing C.5: script_additional_reg_ultra.tcl

```
#*****************************************************************************
# Script used to synthesis the floating point multiplier
#*****************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage2_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_pipeline_reg_stage2.vhd}
#elaborate top entity
elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) ***********************
# create a clock signal to constraint the sequential paths
set WCP 0.7
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
#compile
compile_ultra
#timing and area report
report_timing > timing_additional_reg_ultra.rpt
report_area > area_additioanl_reg_ultra.rpt
#optimize register command
optimize_register
#timing and area report after the reg optimization
report_timing > timing_additional_reg_ultra_retiming.rpt
```

```
report_area > area_additional_reg_ultra_retiming.rpt
```

Listing C.6: script_MBE.tcl

```
#*****************************************************************************
# Script used to synthesis the floating point multiplier
#*****************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/pack_mult32_MBE_dadda.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/mult32_MBE_dadda.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage2_struct_MBE.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/src/fpmul_pipeline_reg.vhd}
#elaborate top entity
elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) ***********************
# create a clock signal to constraint the sequential paths
set WCP 1.6
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
#compile
compile_ultra
#timing and area report
report_timing > timing_report_MBE.rpt
report_area > area_report_MBE.rpt
```

Listing C.7: script_MBE_additional_reg.rpt

```
#*****************************************************************************
# This script is used to synthesize the pipeline multiplier
#*****************************************************************************
#preserve name netlist
set power_preserve_rtl_hier_name true
#list all possible file used by the block under synthesis
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/reg.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ unpackfp_unpackfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/packfp_packfp.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/fpround_fpround.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ fpnormalize_fpnormalize.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ pack_mult32_MBE_dadda.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/mult32_MBE_dadda.vhd
    ↪ }
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ fpmul_stage4_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ fpmul_stage3_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ fpmul_stage2_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/
    ↪ fpmul_stage1_struct.vhd}
analyze -library WORK -format vhdl {/home/isa25/Desktop/lab2/syn2_3B/test2_3/fpmul_pipeline.vhd}
```

```
#elaborate top entity
elaborate FPmul -architecture pipeline -library WORK
#****************** CONSTRAINT THE SYNTHESIS (timing) ***********************
# create a clock signal to constraint the sequential paths
set WCP 0.7
create_clock -name "CLOCK" -period $WCP clk
set_dont_touch_network CLOCK
#clock uncertainty
set_clock_uncertainty 0.07 [get_clocks CLOCK]
ungroup -all -flatten
#compile
compile_ultra
#timing and area report
report_timing > timing_report_MBE_BORU.rpt
report_area -hierarchy > area_report_MBE_BORU.rpt
#
optimize_register
#timing and area report
report_timing > timing_report_MBE_AORU.rpt
report_area -hierarchy > area_report_MBE_AORU.rpt
```

APPENDIX D

Synthesis report listings

Listing D.1: resource_no_impl.rpt

```
****************************************
Report : resources
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 17:56:08 2019
****************************************

Resource Sharing Report for design FPmul in file
        /home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd

===============================================================================
|          |              |            | Contained     |                      |
| Resource | Module       | Parameters | Resources     | Contained Operations |
===============================================================================
| r343     | DW01_add     | width=8    |               | add_1_root_I2/add_126_2 |
| r347     | DW01_inc     | width=25   |               | I3/I11/add_45        |
| r349     | DW01_add     | width=8    |               | I3/I9/add_41_aco     |
| r351     | DW01_add     | width=8    |               | I4/I1/add_41_aco     |
| r931     | DW_mult_uns  | a_width=32 |               | I2/mult_134          |
|          |              | b_width=32 |               |                      |
===============================================================================


Implementation Report
===============================================================================
|                    |                  | Current            | Set            |
| Cell               | Module           | Implementation     | Implementation |
===============================================================================
| I2/mult_134        | DW_mult_uns      | pparch (area,speed)                 |
|                    |                  | mult_arch: benc_radix8              |
| add_1_root_I2/add_126_2               |                    |                |
|                    | DW01_add         | rpl                |                |
| I3/I11/add_45      | DW01_inc         | rpl                |                |
===============================================================================

1
```

Listing D.2: area_no_impl.rpt

```
****************************************
Report : area
Design : FPmul
```

```
Version: O-2018.06-SP4
Date   : Mon Dec  2 17:49:37 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                          301
Number of nets:                          2634
Number of cells:                         2201
Number of combinational cells:           1937
Number of sequential cells:               243
Number of macros/black boxes:               0
Number of buf/inv:                        322
Number of references:                      28

Combinational area:               2861.894003
Buf/Inv area:                      232.217999
Noncombinational area:            1137.149965
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  3999.043968
Total area:                 undefined
1
```

Listing D.3: timing_no_impl.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 17:49:37 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I1/A_SIG_reg[8]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[27]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                                 Incr       Path
  -----------------------------------------------------------------------
  clock CLOCK (rise edge)                               0.00       0.00
  clock network delay (ideal)                           0.00       0.00
  I1/A_SIG_reg[8]/CK (DFF_X1)                           0.00       0.00 r
  I1/A_SIG_reg[8]/Q (DFF_X1)                            0.08       0.08 f
  I2/mult_134/a[8] (FPmul_DW_mult_uns_1)                0.00       0.08 f
  I2/mult_134/U3244/ZN (INV_X2)                         0.06       0.14 r
  I2/mult_134/U2556/ZN (INV_X1)                         0.03       0.17 f
  I2/mult_134/U2527/ZN (XNOR2_X1)                       0.06       0.23 f
  I2/mult_134/U2753/ZN (AND3_X1)                        0.05       0.29 f
  I2/mult_134/U1762/Z (BUF_X4)                          0.06       0.35 f
  I2/mult_134/U2408/ZN (AOI222_X1)                      0.12       0.47 r
  I2/mult_134/U2885/ZN (OAI21_X1)                         0.04       0.51 f
  I2/mult_134/U2407/Z (XOR2_X1)                           0.07       0.58 f
  I2/mult_134/U644/CO (HA_X1)                             0.06       0.64 f
  I2/mult_134/U640/CO (FA_X1)                             0.09       0.73 f
  I2/mult_134/U635/S (FA_X1)                              0.13       0.86 r
  I2/mult_134/U634/S (FA_X1)                              0.11       0.98 f
  I2/mult_134/U2137/ZN (AND2_X1)                          0.04       1.02 f
  I2/mult_134/U2746/ZN (AOI21_X1)                         0.04       1.06 r
  I2/mult_134/U2745/ZN (OAI21_X1)                         0.03       1.09 f
  I2/mult_134/U3064/ZN (AOI21_X1)                         0.04       1.14 r
  I2/mult_134/U2636/ZN (OAI21_X1)                         0.03       1.17 f
  I2/mult_134/U2011/ZN (AND2_X1)                          0.04       1.21 f
  I2/mult_134/U1957/ZN (NOR2_X1)                          0.05       1.25 r
  I2/mult_134/U3038/ZN (OAI21_X1)                         0.04       1.29 f
  I2/mult_134/U2006/ZN (AOI21_X1)                         0.05       1.35 r
  I2/mult_134/U3212/ZN (OAI21_X1)                         0.04       1.38 f
  I2/mult_134/U3211/ZN (AOI21_X1)                         0.04       1.43 r
  I2/mult_134/U2040/ZN (XNOR2_X1)                         0.06       1.49 r
  I2/mult_134/product[47] (FPmul_DW_mult_uns_1)           0.00       1.49 r
  I2/SIG_in_reg[27]/D (DFF_X1)                            0.01       1.50 r
  data arrival time                                                  1.50

  clock CLOCK (rise edge)                                 1.60       1.60
  clock network delay (ideal)                             0.00       1.60
  clock uncertainty                                      -0.07       1.53
  I2/SIG_in_reg[27]/CK (DFF_X1)                           0.00       1.53 r
  library setup time                                     -0.03       1.50
  data required time                                                 1.50
  --------------------------------------------------------------------------
  data required time                                                 1.50
  data arrival time                                                 -1.50
  --------------------------------------------------------------------------
  slack (MET)                                                        0.00


1
```

Listing D.4: resource_CSA.rpt

```
rpt
****************************************
Report : resources
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 18:06:18 2019
****************************************

Resource Sharing Report for design FPmul in file
        /home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd

==============================================================================
|          |              |            | Contained     |                     |
| Resource | Module       | Parameters | Resources     | Contained Operations |
==============================================================================
| r119     | DW01_add     | width=8    |               | add_1_root_I2/add_126_2 |
| r121     | DW02_mult    | A_width=32 |               | I2/mult_134          |
|          |              | B_width=32 |               |                      |
| r123     | DW01_inc     | width=25   |               | I3/I11/add_45        |
| r125     | DW01_add     | width=8    |               | I3/I9/add_41_aco     |
| r127     | DW01_add     | width=8    |               | I4/I1/add_41_aco     |
==============================================================================


Implementation Report
==============================================================================
|                    |                  | Current            | Set            |
| Cell               | Module           | Implementation     | Implementation |
==============================================================================
| add_1_root_I2/add_126_2               |                    |                |
|                    | DW01_add          | rpl                  |                  |
| I2/mult_134        | DW02_mult         | csa                  | csa              |
| I3/I11/add_45      | DW01_inc          | rpl                  |                  |
==============================================================================

1
```

Listing D.5: area_CSA.rpt

```
****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 18:56:43 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
         ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                          490
Number of nets:                          3095
Number of cells:                         2262
Number of combinational cells:           1991
Number of sequential cells:               243
Number of macros/black boxes:               0
Number of buf/inv:                        217
Number of references:                      25

Combinational area:               3710.966004
Buf/Inv area:                      153.215998
Noncombinational area:            1140.341965
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  4851.307970
Total area:                 undefined
1
```

Listing D.6: timing_CSA.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 18:56:42 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I1/A_SIG_reg[0]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[27]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
```

```
-----------------------------------------------
FPmul             5K_hvratio_1_1       NangateOpenCellLibrary

Point                                                  Incr       Path
---------------------------------------------------------------------------
clock CLOCK (rise edge)                                0.00       0.00
clock network delay (ideal)                            0.00       0.00
I1/A_SIG_reg[0]/CK (SDFF_X1)                           0.00       0.00 r
I1/A_SIG_reg[0]/Q (SDFF_X1)                            0.07       0.07 f
I2/mult_134/A[0] (FPmul_DW02_mult_0)                   0.00       0.07 f
I2/mult_134/U586/ZN (INV_X1)                           0.05       0.11 r
I2/mult_134/U1144/ZN (NOR2_X1)                         0.03       0.15 f
I2/mult_134/S0_21/S (FA_X1)                            0.13       0.28 r
I2/mult_134/S2_2_20/S (FA_X1)                          0.11       0.39 f
I2/mult_134/S2_3_19/S (FA_X1)                          0.13       0.53 r
I2/mult_134/S2_4_18/S (FA_X1)                          0.11       0.64 f
I2/mult_134/S2_5_17/S (FA_X1)                          0.13       0.77 r
I2/mult_134/S2_6_16/S (FA_X1)                          0.11       0.89 f
I2/mult_134/S2_7_15/S (FA_X1)                          0.13       1.02 r
I2/mult_134/S2_8_14/S (FA_X1)                          0.11       1.13 f
I2/mult_134/S2_9_13/S (FA_X1)                          0.13       1.27 r
I2/mult_134/S2_10_12/S (FA_X1)                         0.11       1.38 f
I2/mult_134/S2_11_11/S (FA_X1)                         0.13       1.52 r
I2/mult_134/S2_12_10/S (FA_X1)                         0.11       1.63 f
I2/mult_134/S2_13_9/CO (FA_X1)                         0.09       1.72 f
I2/mult_134/S2_14_9/CO (FA_X1)                         0.11       1.83 f
I2/mult_134/S2_15_9/CO (FA_X1)                         0.11       1.93 f
I2/mult_134/S2_16_9/CO (FA_X1)                         0.11       2.04 f
I2/mult_134/S2_17_9/CO (FA_X1)                         0.11       2.14 f
I2/mult_134/S2_18_9/CO (FA_X1)                         0.11       2.25 f
I2/mult_134/S2_19_9/CO (FA_X1)                         0.11       2.36 f
I2/mult_134/S2_20_9/CO (FA_X1)                         0.11       2.46 f
I2/mult_134/S2_21_9/CO (FA_X1)                         0.11       2.57 f
I2/mult_134/S2_22_9/CO (FA_X1)                         0.11       2.67 f
I2/mult_134/S2_23_9/S (FA_X1)                          0.14       2.81 f
I2/mult_134/U400/Z (XOR2_X1)                           0.08       2.89 f
I2/mult_134/U130/ZN (AND2_X1)                          0.04       2.93 f
I2/mult_134/S2_26_7/S (FA_X1)                          0.14       3.07 r
I2/mult_134/U72/ZN (NAND2_X1)                          0.03       3.11 f
I2/mult_134/U2/ZN (INV_X2)                             0.03       3.14 r
I2/mult_134/U505/Z (XOR2_X1)                           0.08       3.22 r
I2/mult_134/U506/Z (XOR2_X1)                           0.08       3.30 r
I2/mult_134/U507/Z (XOR2_X1)                           0.08       3.38 r
I2/mult_134/U503/Z (XOR2_X1)                           0.08       3.46 r
I2/mult_134/U515/Z (XOR2_X1)                           0.08       3.53 r
I2/mult_134/FS_1/A[32] (FPmul_DW01_add_2)              0.00       3.53 r
I2/mult_134/FS_1/U102/ZN (AND2_X1)                     0.06       3.59 r
I2/mult_134/FS_1/U96/ZN (AOI21_X1)                     0.03       3.62 f
I2/mult_134/FS_1/U94/ZN (OAI21_X1)                     0.04       3.66 r
I2/mult_134/FS_1/U86/ZN (AOI21_X1)                     0.03       3.69 f
I2/mult_134/FS_1/U84/ZN (OAI21_X1)                     0.05       3.74 r
I2/mult_134/FS_1/U76/ZN (AOI21_X1)                     0.03       3.77 f
I2/mult_134/FS_1/U20/ZN (OAI21_X1)                     0.05       3.83 r
I2/mult_134/FS_1/U4/ZN (AOI21_X1)                      0.03       3.86 f
I2/mult_134/FS_1/U65/ZN (OAI21_X1)                     0.04       3.90 r
I2/mult_134/FS_1/U16/ZN (AOI21_X1)                     0.03       3.93 f
I2/mult_134/FS_1/U55/ZN (OAI21_X1)                     0.04       3.97 r
I2/mult_134/FS_1/U48/ZN (AOI21_X1)                     0.03       4.00 f
I2/mult_134/FS_1/U21/ZN (OAI21_X1)                     0.04       4.04 r
I2/mult_134/FS_1/U44/ZN (AOI21_X1)                     0.03       4.07 f
I2/mult_134/FS_1/U30/ZN (XNOR2_X1)                     0.05       4.12 f
I2/mult_134/FS_1/SUM[45] (FPmul_DW01_add_2)            0.00       4.12 f
I2/mult_134/PRODUCT[47] (FPmul_DW02_mult_0)            0.00       4.12 f
I2/SIG_in_reg[27]/D (DFF_X1)                           0.01       4.13 f
data arrival time                                                 4.13

clock CLOCK (rise edge)                                4.25       4.25
clock network delay (ideal)                            0.00       4.25
clock uncertainty                                     -0.07       4.18
I2/SIG_in_reg[27]/CK (DFF_X1)                          0.00       4.18 r
library setup time                                    -0.04       4.14
data required time                                                4.14
```

```
  ---------------------------------------------------------------------------
  data required time                                                   4.14
  data arrival time                                                   -4.13
  ---------------------------------------------------------------------------
  slack (MET)                                                          0.01


1
```

Listing D.7: resource_PPARCH.rpt

```


****************************************
Report : resources
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 18:40:01 2019
****************************************

Resource Sharing Report for design FPmul in file
        /home/isa25/Desktop/lab2/src/fpmul_stage1_struct.vhd

===============================================================================
|          |              |            | Contained     |                      |
| Resource | Module       | Parameters | Resources     | Contained Operations |
===============================================================================
| r120     | DW01_add     | width=8    |               | add_1_root_I2/add_126_2 |
| r122     | DW02_mult    | A_width=32 |               | I2/mult_134          |
|          |              | B_width=32 |               |                      |
| r124     | DW01_inc     | width=25   |               | I3/I11/add_45        |
| r126     | DW01_add     | width=8    |               | I3/I9/add_41_aco     |
| r128     | DW01_add     | width=8    |               | I4/I1/add_41_aco     |
===============================================================================


Implementation Report
===============================================================================
|                    |                  | Current            | Set            |
| Cell               | Module           | Implementation     | Implementation |
===============================================================================
| I2/mult_134        | DW02_mult        | pparch (area,speed) | pparch        |
|                    |                  | mult_arch: benc_radix4              |
| I3/I11/add_45      | DW01_inc         | pparch (area,speed)                 |
| add_1_root_I2/add_126_2               |                    |                |
|                    | DW01_add         | rpl                |                |
===============================================================================

1
```

Listing D.8: area_PPARCH.rpt

```


****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 19:00:03 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                          302
Number of nets:                          2619
Number of cells:                        2184
Number of combinational cells:          1919
Number of sequential cells:              243
Number of macros/black boxes:              0
Number of buf/inv:                       310
Number of references:                     27

Combinational area:              2849.924004
Buf/Inv area:                     223.705999
Noncombinational area:           1137.149965
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 3987.073968
Total area:                 undefined
1
```

Listing D.9: timing_PPARCH.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Mon Dec  2 19:00:03 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I1/B_SIG_reg[7]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[23]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                                   Incr       Path
  --------------------------------------------------------------------------
  clock CLOCK (rise edge)                                 0.00       0.00
  clock network delay (ideal)                             0.00       0.00
  I1/B_SIG_reg[7]/CK (DFF_X1)                             0.00       0.00 r
  I1/B_SIG_reg[7]/Q (DFF_X1)                              0.10       0.10 r
  I2/mult_134/B[7] (FPmul_DW02_mult_1)                    0.00       0.10 r
  I2/mult_134/U3103/ZN (NOR2_X1)                          0.04       0.14 f
  I2/mult_134/U2740/ZN (NOR2_X1)                          0.06       0.19 r
  I2/mult_134/U2611/ZN (NAND2_X1)                         0.03       0.22 f
  I2/mult_134/U2511/ZN (OAI21_X1)                         0.07       0.29 r
  I2/mult_134/U2512/ZN (INV_X1)                           0.04       0.33 f
  I2/mult_134/U3217/ZN (OAI21_X1)                         0.05       0.38 r
  I2/mult_134/U2818/ZN (XNOR2_X1)                         0.06       0.44 r
  I2/mult_134/U2582/ZN (INV_X1)                           0.03       0.48 f
  I2/mult_134/U3206/ZN (OAI21_X1)                         0.05       0.52 r
  I2/mult_134/U1757/ZN (XNOR2_X1)                         0.06       0.58 r
  I2/mult_134/U504/S (FA_X1)                              0.12       0.70 f
  I2/mult_134/U502/CO (FA_X1)                             0.09       0.79 f
  I2/mult_134/U496/S (FA_X1)                              0.13       0.93 r
  I2/mult_134/U495/S (FA_X1)                              0.11       1.04 f
  I2/mult_134/U1937/ZN (OR2_X1)                           0.06       1.10 f
  I2/mult_134/U2735/ZN (NAND2_X1)                         0.04       1.15 r
  I2/mult_134/U3202/ZN (OAI21_X1)                         0.03       1.18 f
  I2/mult_134/U3111/ZN (AOI21_X1)                         0.05       1.23 r
  I2/mult_134/U3174/ZN (OAI21_X1)                         0.04       1.27 f
  I2/mult_134/U1930/ZN (AOI21_X1)                         0.07       1.34 r
  I2/mult_134/U2466/ZN (INV_X1)                           0.04       1.38 f
  I2/mult_134/U3160/ZN (AOI21_X1)                         0.05       1.43 r
  I2/mult_134/U2012/ZN (XNOR2_X1)                         0.06       1.49 r
  I2/mult_134/PRODUCT[43] (FPmul_DW02_mult_1)             0.00       1.49 r
  I2/SIG_in_reg[23]/D (DFF_X1)                            0.01       1.50 r
  data arrival time                                                  1.50

  clock CLOCK (rise edge)                                 1.60       1.60
  clock network delay (ideal)                             0.00       1.60
  clock uncertainty                                      -0.07       1.53
  I2/SIG_in_reg[23]/CK (DFF_X1)                           0.00       1.53 r
  library setup time                                     -0.03       1.50
  data required time                                                 1.50
  --------------------------------------------------------------------------
  data required time                                                 1.50
  data arrival time                                                 -1.50
  --------------------------------------------------------------------------
  slack (MET)                                                        0.00


1
```

Listing D.10: area_additional_reg.rpt

```
****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 12:19:27 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                          301
Number of nets:                          2696
Number of cells:                         2179
Number of combinational cells:           1874
Number of sequential cells:               283
Number of macros/black boxes:               0
Number of buf/inv:                        354
Number of references:                      27

Combinational area:               2973.614009
Buf/Inv area:                      231.420000
Noncombinational area:            1326.541959
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  4300.155969
Total area:                 undefined
1
```

Listing D.11: timing_additional_reg.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 12:19:27 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I1/A_SIG_reg[19]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[21]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                                   Incr       Path
  --------------------------------------------------------------------------
  clock CLOCK (rise edge)                                 0.00       0.00
  clock network delay (ideal)                             0.00       0.00
  I1/A_SIG_reg[19]/CK (DFF_X1)                            0.00       0.00 r
  I1/A_SIG_reg[19]/Q (DFF_X1)                             0.10       0.10 r
  I2/mult_134/a[19] (FPmul_DW_mult_uns_2)                 0.00       0.10 r
  I2/mult_134/U2765/ZN (XNOR2_X1)                         0.07       0.17 r
  I2/mult_134/U1646/Z (BUF_X1)                            0.04       0.21 r
  I2/mult_134/U1667/ZN (NAND2_X1)                         0.05       0.26 f
  I2/mult_134/U2578/ZN (OAI22_X1)                         0.06       0.32 r
  I2/mult_134/U645/S (FA_X1)                              0.13       0.45 f
  I2/mult_134/U640/CO (FA_X1)                             0.09       0.54 f
  I2/mult_134/U627/S (FA_X1)                              0.15       0.69 r
  I2/mult_134/U625/S (FA_X1)                              0.12       0.80 f
  I2/mult_134/U624/S (FA_X1)                              0.14       0.94 r
  I2/mult_134/U1693/ZN (NOR2_X1)                          0.03       0.97 f
  I2/mult_134/U2511/ZN (NOR2_X1)                          0.05       1.02 r
  I2/mult_134/U2681/ZN (NAND2_X1)                         0.04       1.05 f
  I2/mult_134/U2506/ZN (OAI21_X1)                         0.08       1.14 r
  I2/mult_134/U2523/ZN (AOI21_X1)                         0.05       1.19 f
  I2/mult_134/U2858/ZN (OAI21_X1)                         0.05       1.24 r
  I2/mult_134/U2608/ZN (XNOR2_X1)                         0.06       1.29 r
  I2/mult_134/product[41] (FPmul_DW_mult_uns_2)           0.00       1.29 r
  I2/SIG_in_reg[21]/D (DFF_X1)                            0.01       1.30 r
  data arrival time                                                  1.30

  clock CLOCK (rise edge)                                 0.70       0.70
  clock network delay (ideal)                             0.00       0.70
  clock uncertainty                                      -0.07       0.63
  I2/SIG_in_reg[21]/CK (DFF_X1)                           0.00       0.63 r
  library setup time                                     -0.03       0.60
  data required time                                                 0.60
  --------------------------------------------------------------------------
  data required time                                                 0.60
  data arrival time                                                 -1.30
  --------------------------------------------------------------------------
  slack (VIOLATED)                                                  -0.71


1
```

Listing D.12: area_additional_reg_retiming.rpt

```
****************************************
Report : area
Design : FPmul
```

```
Version: O-2018.06-SP4
Date   : Fri Dec 13 12:22:05 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                          277
Number of nets:                          2922
Number of cells:                         2372
Number of combinational cells:           1807
Number of sequential cells:               546
Number of macros/black boxes:               0
Number of buf/inv:                        221
Number of references:                      26

Combinational area:               2933.714008
Buf/Inv area:                      128.478000
Noncombinational area:            2472.735911
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  5406.449919
Total area:                 undefined
1
```

Listing D.13: timing_additional_reg_retiming.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 12:22:05 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I2/mult_134/CLOCK_r_REG408_S1
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/mult_134/CLOCK_r_REG219_S2
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                                   Incr       Path
  -----------------------------------------------------------------------
  clock CLOCK (rise edge)                                 0.00       0.00
  clock network delay (ideal)                             0.00       0.00
  I2/mult_134/CLOCK_r_REG408_S1/CK (DFF_X1)               0.00       0.00 r
  I2/mult_134/CLOCK_r_REG408_S1/Q (DFF_X1)                0.14       0.14 r
  I2/mult_134/U2182/ZN (XNOR2_X1)                         0.08       0.22 r
  I2/mult_134/U2717/ZN (OAI22_X1)                         0.04       0.26 f
  I2/mult_134/U598/CO (FA_X1)                             0.09       0.36 f
  I2/mult_134/U586/CO (FA_X1)                             0.10       0.46 f
  I2/mult_134/U574/S (FA_X1)                              0.13       0.59 r
  I2/mult_134/CLOCK_r_REG219_S2/D (DFF_X1)                0.01       0.60 r
  data arrival time                                                  0.60
  clock CLOCK (rise edge)                                 0.70       0.70
  clock network delay (ideal)                             0.00       0.70
  clock uncertainty                                      -0.07       0.63
  I2/mult_134/CLOCK_r_REG219_S2/CK (DFF_X1)               0.00       0.63 r
  library setup time                                     -0.03       0.60
  data required time                                                 0.60
  --------------------------------------------------------------------------
  data required time                                                 0.60
  data arrival time                                                 -0.60
  --------------------------------------------------------------------------
  slack (MET)                                                        0.00


1
```

Listing D.14: area_additional_reg_ultra.rpt

```
****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Thu Dec 12 14:57:40 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
         ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                           97
Number of nets:                          2824
Number of cells:                         2514
Number of combinational cells:           2227
Number of sequential cells:               283
Number of macros/black boxes:               0
Number of buf/inv:                        293
Number of references:                      46

Combinational area:               3122.041990
Buf/Inv area:                      197.638000
Noncombinational area:            1303.399957
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  4425.441947
Total area:                 undefined
1
```

Listing D.15: timing_additional_reg_ultra.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Thu Dec 12 14:57:40 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top
```

```
  Startpoint: I1/A_SIG_reg[17]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[15]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock CLOCK (rise edge)                  0.00       0.00
  clock network delay (ideal)              0.00       0.00
  I1/A_SIG_reg[17]/CK (DFF_X1)             0.00       0.00 r
  I1/A_SIG_reg[17]/QN (DFF_X1)             0.08       0.08 r
  U363/ZN (XNOR2_X1)                       0.07       0.15 r
  U1185/ZN (NAND2_X1)                      0.04       0.19 f
  U1202/Z (BUF_X2)                         0.04       0.23 f
  U1434/ZN (OAI22_X1)                      0.07       0.30 r
  U1436/ZN (XNOR2_X1)                      0.07       0.37 r
  U1438/ZN (XNOR2_X1)                      0.04       0.41 f
  U1481/CO (FA_X1)                         0.11       0.51 f
  U1781/S (FA_X1)                          0.12       0.64 f
  U1774/CO (FA_X1)                         0.11       0.75 f
  U1777/ZN (XNOR2_X1)                      0.06       0.81 f
  U1787/ZN (XNOR2_X1)                      0.06       0.87 f
  U1813/ZN (NAND2_X1)                      0.03       0.91 r
  U1815/ZN (OAI21_X1)                      0.03       0.94 f
  U1819/ZN (AOI21_X1)                      0.05       0.99 r
  U1827/ZN (OAI21_X1)                      0.03       1.02 f
  U612/ZN (AOI21_X1)                       0.06       1.08 r
  U611/Z (BUF_X2)                          0.06       1.14 r
  U2153/ZN (OAI21_X1)                      0.04       1.18 f
  U2155/ZN (XNOR2_X1)                      0.05       1.23 f
  I2/SIG_in_reg[15]/D (DFF_X1)             0.01       1.24 f
  data arrival time                                   1.24

  clock CLOCK (rise edge)                  0.70       0.70
  clock network delay (ideal)              0.00       0.70
  clock uncertainty                       -0.07       0.63
  I2/SIG_in_reg[15]/CK (DFF_X1)            0.00       0.63 r
  library setup time                      -0.04       0.59
  data required time                                  0.59
  -----------------------------------------------------------
  data required time                                  0.59
  data arrival time                                  -1.24
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.65


1
```

Listing D.16: area_additional_reg_ultra_retiming.rpt

```
****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Thu Dec 12 14:58:27 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)
```

```
Number of ports:                          97
Number of nets:                         3016
Number of cells:                        2681
Number of combinational cells:          2208
Number of sequential cells:              473
Number of macros/black boxes:              0
Number of buf/inv:                       213
Number of references:                     36

Combinational area:              3089.057988
Buf/Inv area:                     125.286000
Noncombinational area:           2156.461925
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 5245.519913
Total area:                 undefined
1
```

Listing D.17: timing_additional_reg_ultra_retiming.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Thu Dec 12 14:58:27 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: CLOCK_r_REG462_S1
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: CLOCK_r_REG421_S2
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock CLOCK (rise edge)                  0.00       0.00
  clock network delay (ideal)              0.00       0.00
  CLOCK_r_REG462_S1/CK (DFF_X2)            0.00       0.00 r
  CLOCK_r_REG462_S1/Q (DFF_X2)             0.14       0.14 r
  U827/ZN (XNOR2_X1)                       0.08       0.22 r
  U854/ZN (OAI22_X1)                       0.04       0.27 f
  U913/S (FA_X1)                           0.15       0.42 r
  U919/S (FA_X1)                           0.11       0.53 f
  U915/ZN (OR2_X1)                         0.05       0.58 f
  CLOCK_r_REG421_S2/D (DFF_X1)             0.01       0.59 f
  data arrival time                                   0.59

  clock CLOCK (rise edge)                  0.70       0.70
  clock network delay (ideal)              0.00       0.70
  clock uncertainty                       -0.07       0.63
  CLOCK_r_REG421_S2/CK (DFF_X1)            0.00       0.63 r
  library setup time                      -0.04       0.59
  data required time                                  0.59
  -----------------------------------------------------------
  data required time                                0.59
  data arrival time                                -0.59
  -----------------------------------------------------------
  slack (MET)                                       0.00


1
```

Listing D.18: area_MBE.rpt

```


****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 02:54:23 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                           97
Number of nets:                          3937
Number of cells:                         3651
Number of combinational cells:           3401
Number of sequential cells:               243
Number of macros/black boxes:               0
Number of buf/inv:                        618
Number of references:                      42

Combinational area:               3997.979995
Buf/Inv area:                      370.006001
Noncombinational area:            1116.933961
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  5114.913956
Total area:                 undefined
1
```

Listing D.19: timing_MBE.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 02:54:23 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I1/B_SIG_reg[13]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[22]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max
  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock CLOCK (rise edge)                  0.00       0.00
  clock network delay (ideal)              0.00       0.00
  I1/B_SIG_reg[13]/CK (DFF_X1)             0.00       0.00 r
  I1/B_SIG_reg[13]/Q (DFF_X1)              0.10       0.10 r
  U1680/Z (BUF_X1)                         0.07       0.17 r
  U2405/ZN (NOR2_X1)                       0.04       0.20 f
  U1663/Z (BUF_X1)                         0.07       0.27 f
  U2221/ZN (AND2_X1)                       0.05       0.33 f
  U2225/ZN (NOR2_X1)                       0.04       0.37 r
  U4032/ZN (NAND2_X1)                      0.03       0.40 f
  U4036/ZN (NAND2_X1)                      0.04       0.44 r
  U4037/ZN (OAI21_X1)                      0.04       0.48 f
  intadd_53/U3/S (FA_X1)                   0.15       0.62 r
  U2524/ZN (INV_X1)                        0.03       0.65 f
  U2643/ZN (NAND2_X1)                      0.03       0.68 r
  U2645/ZN (NAND3_X1)                      0.05       0.73 f
  U2819/ZN (XNOR2_X1)                      0.07       0.79 f
  U2820/ZN (XNOR2_X1)                      0.06       0.85 f
  intadd_0/U3/CO (FA_X1)                   0.09       0.95 f
  intadd_0/U2/S (FA_X1)                    0.13       1.08 f
  U3593/ZN (NAND2_X1)                      0.04       1.12 r
  U3594/ZN (OAI21_X1)                      0.03       1.16 f
  U3595/ZN (AOI21_X1)                      0.06       1.22 r
  U3608/ZN (OAI21_X1)                      0.03       1.25 f
  U2188/ZN (AOI21_X1)                      0.06       1.31 r
  U3653/Z (BUF_X1)                         0.04       1.35 r
  U2164/ZN (OAI21_X1)                      0.03       1.38 f
  U2077/ZN (AOI21_X1)                      0.04       1.43 r
  U2154/ZN (XNOR2_X1)                      0.06       1.49 r
  I2/SIG_in_reg[22]/D (DFF_X1)             0.01       1.50 r
  data arrival time                                   1.50

  clock CLOCK (rise edge)                  1.60       1.60
  clock network delay (ideal)              0.00       1.60
  clock uncertainty                       -0.07       1.53
  I2/SIG_in_reg[22]/CK (DFF_X1)            0.00       1.53 r
  library setup time                      -0.03       1.50
  data required time                                  1.50
  -----------------------------------------------------------
  data required time                                  1.50
  data arrival time                                  -1.50
  -----------------------------------------------------------
  slack (MET)                                         0.00


1
```

Listing D.20: area_MBE_additional_reg.rpt

```


****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 20:11:15 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                          97
Number of nets:                          4642
Number of cells:                         4415
Number of combinational cells:           4130
Number of sequential cells:               283
Number of macros/black boxes:               0
Number of buf/inv:                        683
Number of references:                      42

Combinational area:               4765.389960
Buf/Inv area:                      411.768001
Noncombinational area:            1320.689959
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  6086.079919
Total area:                 undefined

Hierarchical area distribution
------------------------------

                                  Global cell area          Local cell area
                                  ------------------  ----------------------------
Hierarchical cell                 Absolute    Percent  Combi-     Noncombi-  Black-
                                  Total       Total    national   national   boxes   Design
--------------------------------  ---------   -------  ---------  ---------  ------  ---------
FPmul                             6086.0799     100.0  4765.3900  1320.6900  0.0000  FPmul
--------------------------------  ---------   -------  ---------  ---------  ------  ---------
Total                                                  4765.3900  1320.6900  0.0000

1
```

Listing D.21: timing_MBE_additional_reg.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 20:11:14 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: I1/B_SIG_reg[11]
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: I2/SIG_in_reg[24]
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock CLOCK (rise edge)                  0.00       0.00
  clock network delay (ideal)              0.00       0.00
  I1/B_SIG_reg[11]/CK (DFF_X1)             0.00       0.00 r
  I1/B_SIG_reg[11]/Q (DFF_X1)              0.11       0.11 r
  U2323/ZN (NAND2_X2)                      0.09       0.20 f
  U2485/Z (MUX2_X1)                        0.09       0.29 f
  U2489/ZN (NAND2_X1)                      0.04       0.33 r
  U2615/ZN (XNOR2_X1)                      0.06       0.39 r
  U2617/ZN (XNOR2_X1)                      0.06       0.46 r
  U2641/S (FA_X1)                          0.13       0.58 f
  U2637/CO (FA_X1)                         0.11       0.69 f
  U2661/ZN (INV_X1)                        0.03       0.72 r
  U2662/ZN (OR2_X1)                        0.03       0.76 r
  U2663/ZN (NAND2_X1)                      0.03       0.79 f
  U1738/ZN (OR2_X2)                        0.06       0.84 f
  U2682/ZN (NAND2_X1)                      0.03       0.87 r
  U2683/ZN (XNOR2_X1)                      0.06       0.94 r
  U2777/ZN (XNOR2_X1)                      0.07       1.00 r
  U2867/ZN (NOR2_X1)                       0.03       1.03 f
  U2872/ZN (NOR2_X1)                       0.04       1.08 r
  U2873/ZN (NAND2_X1)                      0.04       1.11 f
  U4955/ZN (OAI21_X2)                      0.11       1.22 r
  U5350/ZN (AOI21_X1)                      0.04       1.26 f
  U5351/ZN (OAI21_X1)                      0.04       1.30 r
  U5353/ZN (XNOR2_X1)                      0.06       1.36 r
  I2/SIG_in_reg[24]/D (DFF_X1)             0.01       1.36 r
  data arrival time                                   1.36

  clock CLOCK (rise edge)                  0.70       0.70
  clock network delay (ideal)              0.00       0.70
  clock uncertainty                       -0.07       0.63
  I2/SIG_in_reg[24]/CK (DFF_X1)            0.00       0.63 r
  library setup time                      -0.03       0.60
  data required time                                  0.60
  -----------------------------------------------------------
  data required time                                  0.60
  data arrival time                                  -1.36
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.77


1
```

Listing D.22: area_MBE_additional_reg_retiming.rpt

```
****************************************
Report : area
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 20:12:01 2019
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /software/dk/nangate45/synopsys/
        ↪ NangateOpenCellLibrary_typical_ecsm_nowlm.db)

Number of ports:                           97
Number of nets:                          4907
Number of cells:                         4615
Number of combinational cells:           4018
Number of sequential cells:               597
Number of macros/black boxes:               0
Number of buf/inv:                        535
Number of references:                      37

Combinational area:               4666.969957
Buf/Inv area:                      305.368001
Noncombinational area:            2703.889903
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  7370.859860
Total area:                 undefined

Hierarchical area distribution
```

```
------------------------------

                                  Global cell area          Local cell area
                                  ------------------  ----------------------------
Hierarchical cell                 Absolute   Percent  Combi-     Noncombi-  Black-
                                  Total      Total    national   national   boxes   Design
--------------------------------  ---------  -------  ---------  ---------  ------  ---------
FPmul                             7370.8599    100.0  4666.9700  2703.8899  0.0000  FPmul
--------------------------------  ---------  -------  ---------  ---------  ------  ---------
Total                                                 4666.9700  2703.8899  0.0000

1
```

Listing D.23: timing_MBE_additional_reg_retiming.rpt

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : FPmul
Version: O-2018.06-SP4
Date   : Fri Dec 13 20:12:01 2019
****************************************

Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

  Startpoint: CLOCK_r_REG589_S1
              (rising edge-triggered flip-flop clocked by CLOCK)
  Endpoint: CLOCK_r_REG422_S2
            (rising edge-triggered flip-flop clocked by CLOCK)
  Path Group: CLOCK
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  FPmul              5K_hvratio_1_1        NangateOpenCellLibrary

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock CLOCK (rise edge)                  0.00       0.00
  clock network delay (ideal)              0.00       0.00
  CLOCK_r_REG589_S1/CK (DFF_X1)            0.00       0.00 r
  CLOCK_r_REG589_S1/Q (DFF_X1)             0.10       0.10 r
  U2125/ZN (AND2_X1)                       0.07       0.16 r
  U2953/ZN (NAND2_X1)                      0.06       0.23 f
  U3514/Z (BUF_X1)                         0.07       0.30 f
  U3652/Z (MUX2_X1)                        0.08       0.37 f
  U3653/ZN (NAND2_X1)                      0.03       0.40 r
  U3654/ZN (NOR2_X1)                       0.03       0.43 f
  U3726/S (FA_X1)                          0.14       0.56 r
  U3727/ZN (INV_X1)                        0.02       0.58 f
  CLOCK_r_REG422_S2/D (DFF_X1)             0.01       0.59 f
  data arrival time                                   0.59

  clock CLOCK (rise edge)                  0.70       0.70
  clock network delay (ideal)              0.00       0.70
  clock uncertainty                       -0.07       0.63
  CLOCK_r_REG422_S2/CK (DFF_X1)            0.00       0.63 r
  library setup time                      -0.04       0.59
  data required time                                  0.59
  -----------------------------------------------------------
  data required time                                  0.59
  data arrival time                                  -0.59
  -----------------------------------------------------------
  slack (MET)                                         0.00
```

```

```