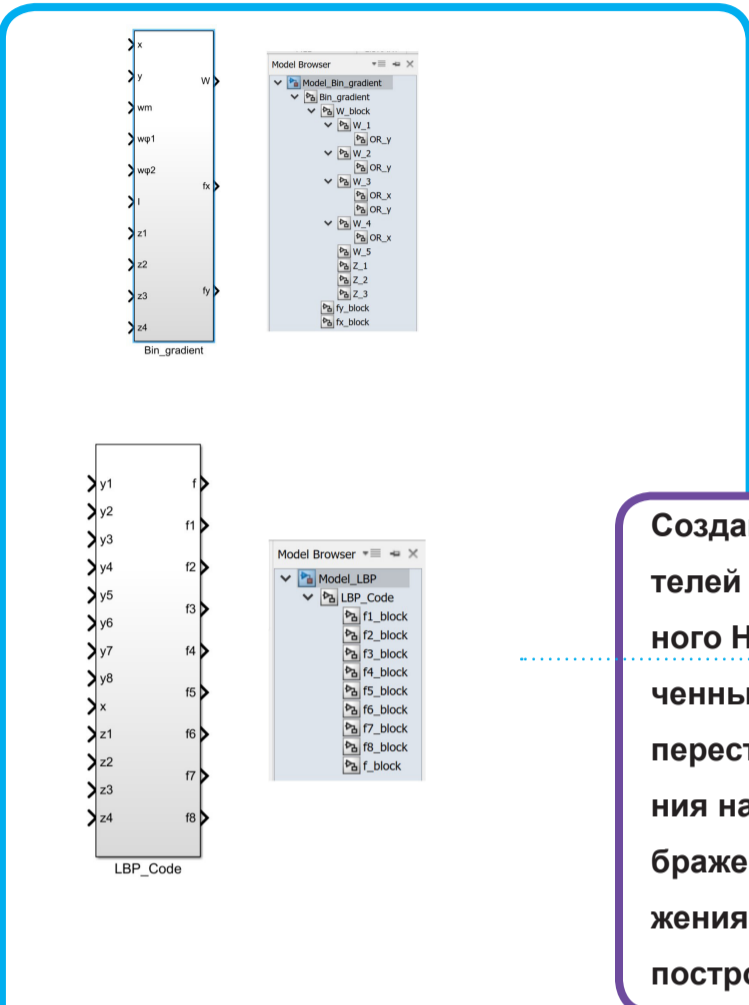


Разработка и исследование реконфигурируемых быстродействующих алгоритмов распознавания изображений для оценки дорожной ситуации на базе специализированных мобильных устройств с параллельно-конвейерной архитектурой

Грант РФФИ 19-29-06078 мк

Авторы: Шидловский С.В., Шашев Д.В., Бондарчук А.С., Гимазов Р.У., Шатравин В., Окунский М.В.



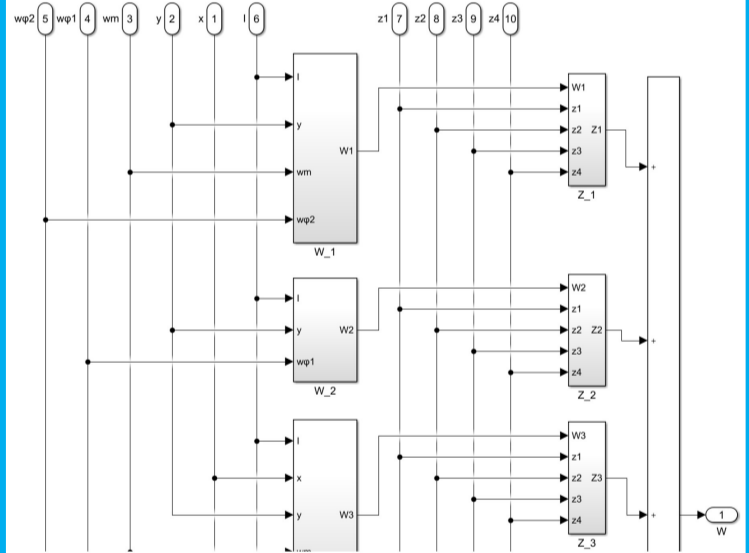
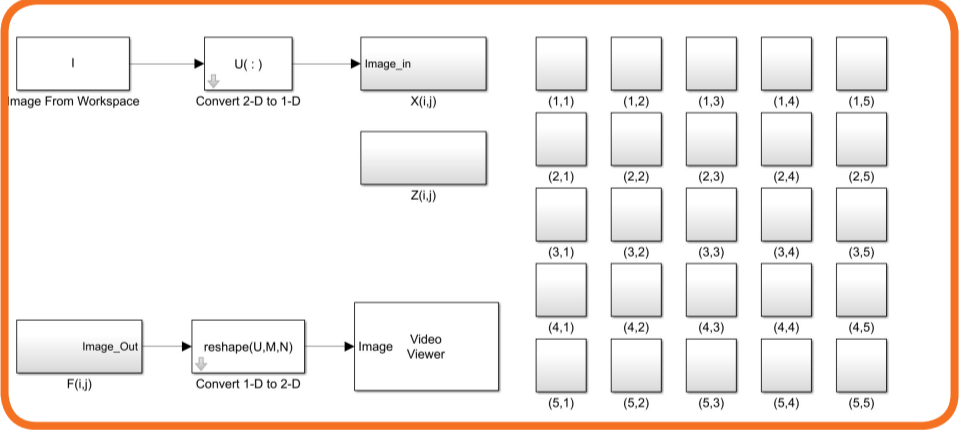
Математическая модель ЭВ при реализации ЛБШ

$$\begin{aligned}
 F_1 &= X \cdot (z_2 z_3 \bar{z}_1 \vee z_2 \bar{z}_1 \bar{z}_4 \vee z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4), \\
 F_2 &= X \cdot (z_2 \bar{z}_1 \vee z_3 z_4 \bar{z}_1 \vee z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4), \\
 F_3 &= X \cdot (z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4 \vee z_2 z_3 \bar{z}_1 \bar{z}_4 \vee z_2 z_4 \bar{z}_1 \bar{z}_3 \vee z_3 z_4 \bar{z}_1 \bar{z}_2), \\
 F_4 &= X \cdot (z_4 \bar{z}_2 \bar{z}_3 \vee z_4 \bar{z}_1 \bar{z}_3 \vee \bar{z}_2 \bar{z}_3 \bar{z}_4 \vee z_2 z_3 \bar{z}_1 \bar{z}_4), \\
 F_5 &= X \cdot (z_4 \bar{z}_2 \bar{z}_3 \vee \bar{z}_2 \bar{z}_3 \bar{z}_4), \\
 F_6 &= X \cdot (\bar{z}_1 \bar{z}_2 \vee \bar{z}_1 \bar{z}_3 \bar{z}_4 \vee \bar{z}_2 \bar{z}_3 \bar{z}_4), \\
 F_7 &= X \cdot (z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4 \vee z_2 \bar{z}_1 \bar{z}_3 \bar{z}_4 \vee z_3 \bar{z}_1 \bar{z}_2 \bar{z}_3 \vee z_4 \bar{z}_1 \bar{z}_2 \bar{z}_3), \\
 F_8 &= X \cdot (z_2 z_3 \bar{z}_1 \vee z_2 \bar{z}_1 \bar{z}_4 \vee z_3 \bar{z}_1 \bar{z}_4 \vee z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4 \vee z_4 \bar{z}_1 \bar{z}_2 \bar{z}_3), \\
 F^j &= M(Y_j, X) \cdot z_1 \bar{z}_2 \bar{z}_3 \bar{z}_4, \text{ where } j = \overline{7,0}.
 \end{aligned}$$

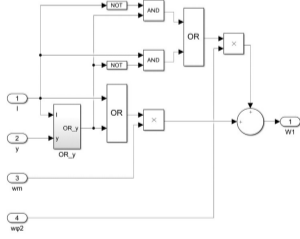
Математическая модель ЭВ при реализации бинарного градиента

$$\begin{cases}
 fx = fy = I, \\
 f = (I \cdot \bar{x} \vee \bar{I} \cdot x) \cdot \bar{z} \vee (I \cdot \bar{y} \vee \bar{I} \cdot y) \cdot z.
 \end{cases}$$

Созданные на первом этапе проекта математические модели элементарных вычислителей (ЭВ) для реализации алгоритмов **ЛБШ** и **бинарного градиента** (модифицированного НОГ) реализованы в качестве **иммитационных моделей** в среде Matlab. Из полученных моделей ЭВ организовывалась матричная структура, отражающая идеологию перестраиваемых вычислительных сред (**ПВС**). На этапе имитационного моделирования на информационные входы ЭВ ПВС подавались значения пикселей тестового изображения. При этом размерность ПВС совпадала с размерностью тестового изображения. По **результатам** имитационного моделирования установлена адекватность построенных моделей ЭВ и ПВС.



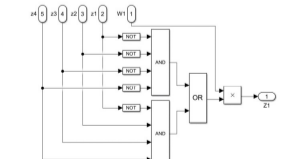
Фрагмент внутренней структуры подсистемы W_block



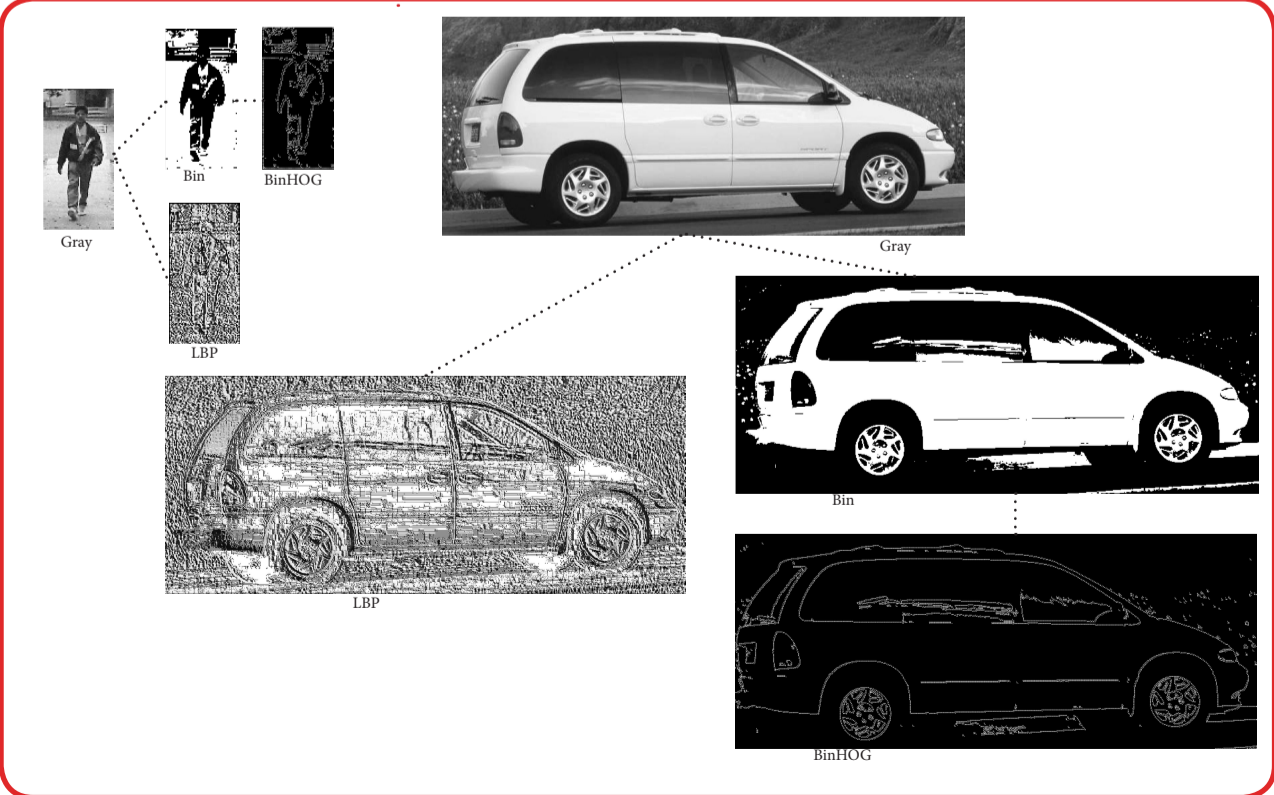
Внутренняя структура подсистемы W_1



Внутренняя структура подсистемы OR_y

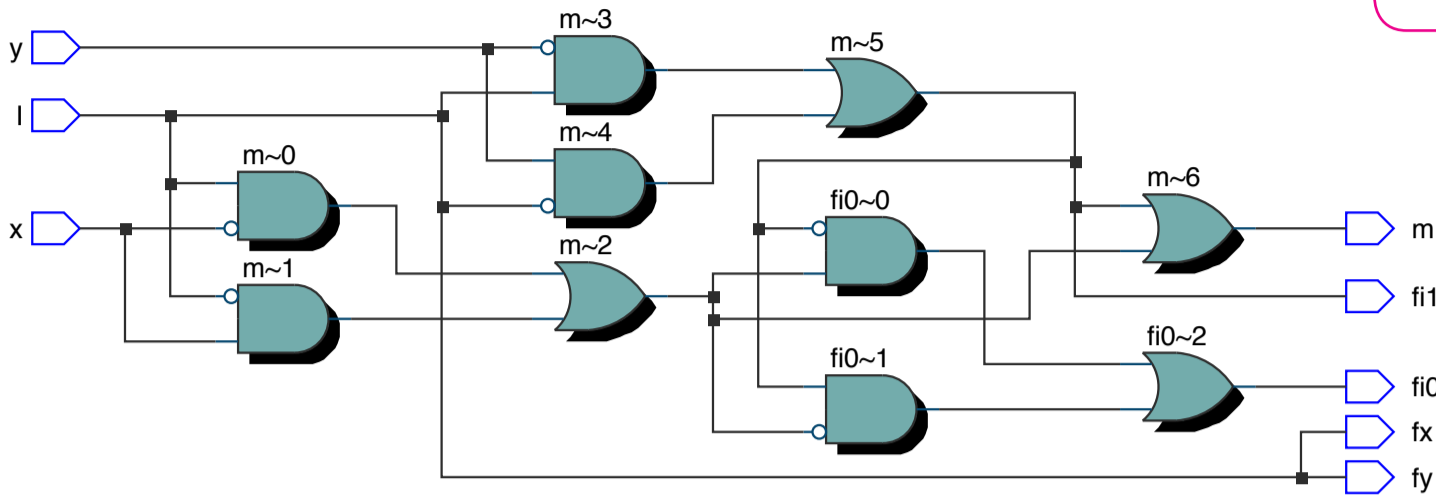


Внутренняя структура подсистемы Z_1



Полученные результаты моделирования позволили перейти к описанию сначала ЭВ, а затем и всей ПВС целиком, на языке SystemVerilog, для последующей реализации предложенной концепции с привлечением технологии ПЛИС (FPGA).

RTL-схема ЭВ, реализующего алгоритм бинарного градиента



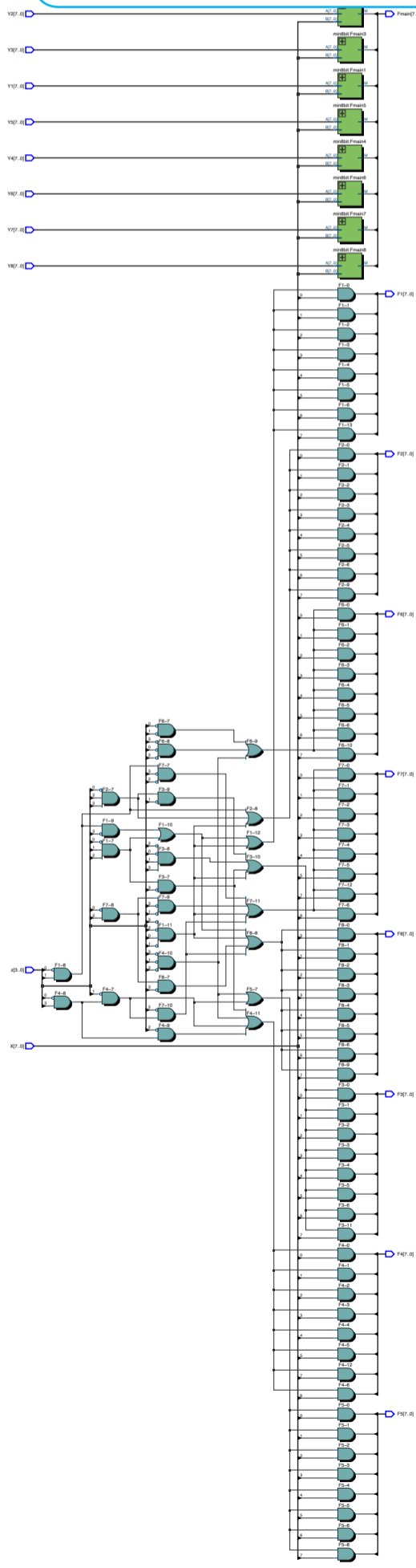
SystemVerilog-код для ЭВ, реализующего алгоритм бинарного градиента

```

1 module bin_hog(
2   input I,x,y,
3   output m, fx, fy, fi1, fi0);
4
5   always_comb
6   begin
7     fx=I;
8     fy=I;
9     m=(I&~x|~I&x)|(I&~y|~I&y);
10    fi1=I&~y|~I&y;
11    fi0=(I&~x|~I&x)&~(I&~y|~I&y)|~(I&~x|~I&x)&(I&~y|~I&y);
12  end
13
14 endmodule

```

RTL-схема ЭВ, реализующего алгоритм ЛБШ



Фрагмент SystemVerilog-код для ЭВ, реализующего алгоритм ЛБШ

```

module lbp(
  input [3:0] z,
  input [7:0] x,
  input [7:0] y1,
  input [7:0] y2,
  input [7:0] y3,
  input [7:0] y4,
  input [7:0] y5,
  input [7:0] y6,
  input [7:0] y7,
  input [7:0] y8,
  output reg [7:0] F1,
  output reg [7:0] F2,
  output reg [7:0] F3,
  output reg [7:0] F4,
  output reg [7:0] F5,
  output reg [7:0] F6,
  output reg [7:0] F7,
  output reg [7:0] F8,
  output reg [7:0] Fmain
);

always_comb
begin
  //==== F1 output =====
  F1[0]=x[0]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[1]=x[1]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[2]=x[2]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[3]=x[3]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[4]=x[4]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[5]=x[5]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[6]=x[6]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];
  F1[7]=x[7]&(z[1]&z[2]&~z[0])|z[1]&~z[0]&~z[3]|z[0]&~z[1]&~z[2]&~z[3];

  //==== F2 output =====
  F2[0]=x[0]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[1]=x[1]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[2]=x[2]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[3]=x[3]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[4]=x[4]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[5]=x[5]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[6]=x[6]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];
  F2[7]=x[7]&(z[1]&z[2]&~z[0])|z[2]&z[3]&~z[0]|z[0]&~z[1]&~z[2]&~z[3];

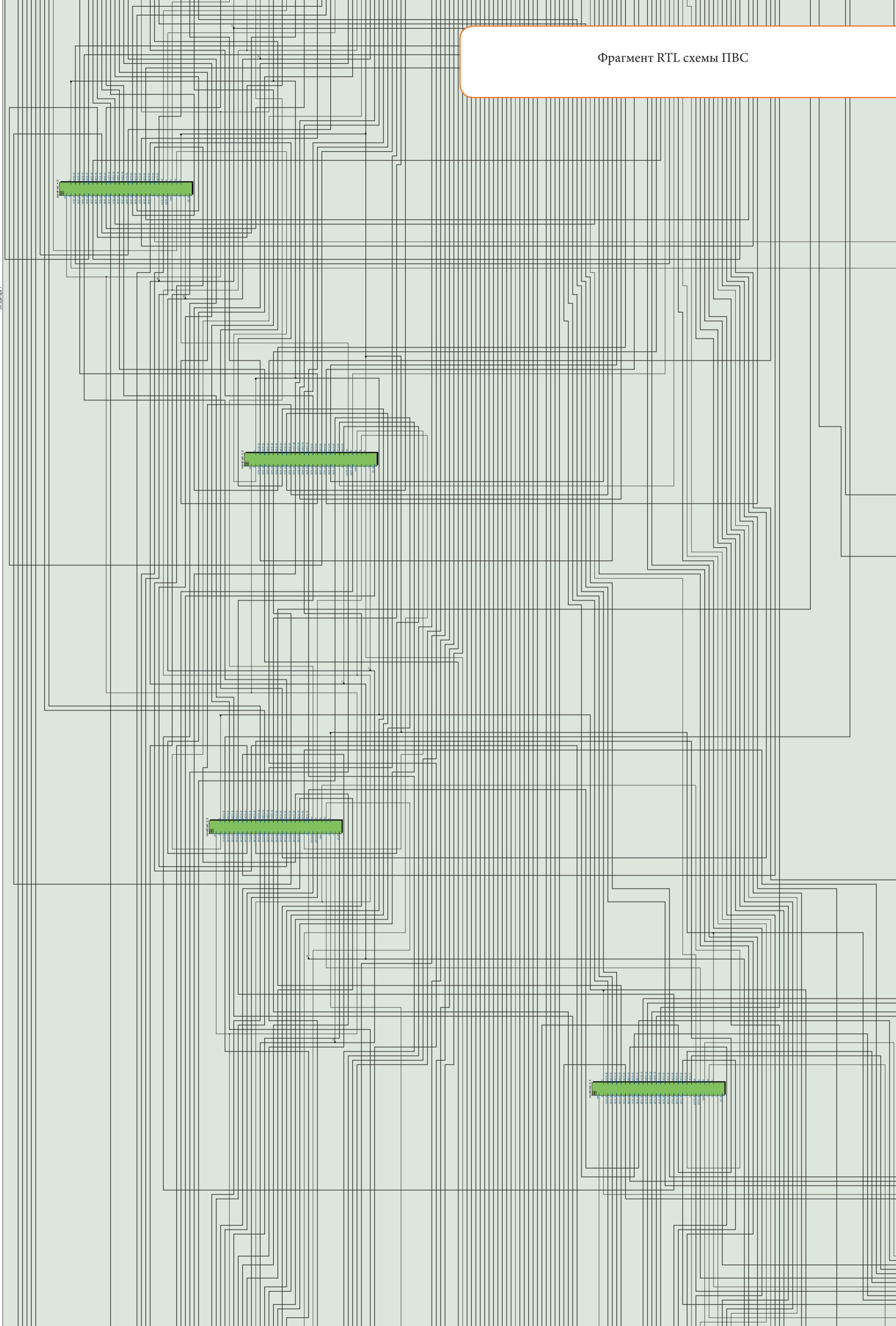
  //==== F3 output =====
  F3[0]=x[0]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[1]=x[1]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[2]=x[2]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[3]=x[3]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[4]=x[4]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[5]=x[5]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[6]=x[6]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];
  F3[7]=x[7]&(z[0]&~z[1]&~z[2]&~z[3])|z[1]&z[2]&~z[0]&~z[3]|z[1]&z[3]&~z[0]&~z[2]|z[2]&z[3]&~z[0]&~z[1];

  //==== F4 output =====
  F4[0]=x[0]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[1]=x[1]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[2]=x[2]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[3]=x[3]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[4]=x[4]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[5]=x[5]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[6]=x[6]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F4[7]=x[7]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];

  //==== F5 output =====
  F5[0]=x[0]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[1]=x[1]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[2]=x[2]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[3]=x[3]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[4]=x[4]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[5]=x[5]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[6]=x[6]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];
  F5[7]=x[7]&(z[3]&~z[0]&~z[1])|z[3]&~z[0]&~z[2]|~z[1]&~z[2]&~z[3]|z[1]&z[2]&~z[0]&~z[3];

```

Фрагмент RTL схемы ПВС



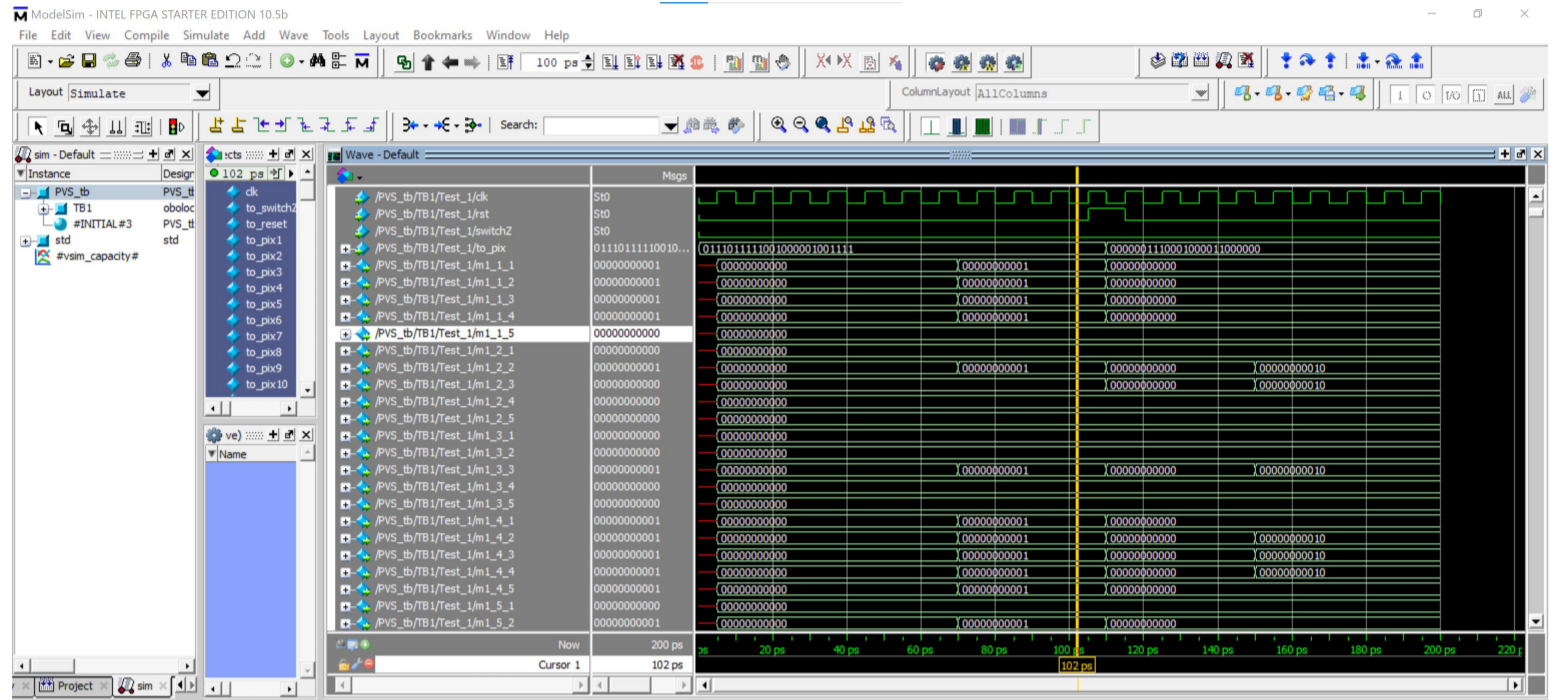

```

PVS_tb.vv
277 initial
278 begin
279   to_reset=1'b0;
280   to_switchZ=1'b0;
281   to_pix1=1'b1;
282   to_pix2=1'b1;
283   to_pix3=1'b1;
284   to_pix4=1'b1;
285   to_pix5=1'b0;
286   to_pix6=1'b0;
287   to_pix7=1'b1;
288   to_pix8=1'b0;
289   to_pix9=1'b0;
290   to_pix10=1'b0;
291   to_pix11=1'b0;
292   to_pix12=1'b0;
293   to_pix13=1'b1;
294   to_pix14=1'b0;
295   to_pix15=1'b0;
296   to_pix16=1'b1;
297   to_pix17=1'b1;
298   to_pix18=1'b1;
299   to_pix19=1'b1;
300   to_pix20=1'b1;
301   to_pix21=1'b0;
302   to_pix22=1'b1;
303   to_pix23=1'b1;
304   to_pix24=1'b1;
305   to_pix25=1'b0;
306   #100
307   @(posedge clk);
308   to_reset=1'b1;
309   @(negedge clk);
310 begin
311   to_pix1=1'b0;
312   to_pix2=1'b0;
313   to_pix3=1'b0;
314   to_pix4=1'b0;
315   to_pix5=1'b0;
316   to_pix6=1'b0;
317   to_pix7=1'b1;
318   to_pix8=1'b1;
319   to_pix9=1'b0;

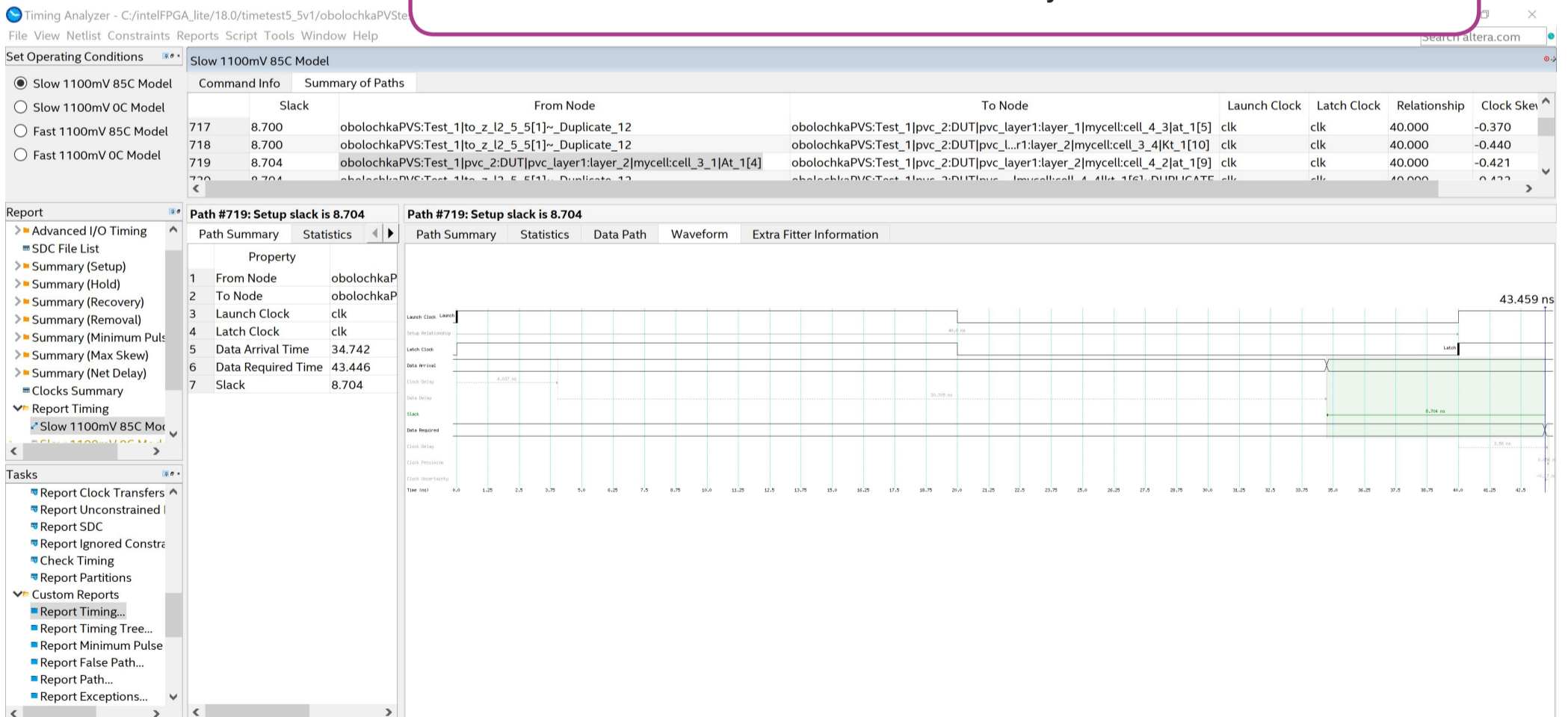
```

Тестирование полученных моделей осуществлялось с применением функциональной и временной симуляции. Функциональная симуляция представляла из себя составления специальных тестирующих сценариев, включающих:

- формирование тестовых бинарных изображений (например для изображения размером 5x5 пикселей тестовый сигнал представлял из себя 25 битный сигнал, каждый бит которого подавался на соответствующий вход ПВС);
- генерацию настроечных кодов каждого ЭВ ПВС.



Временная симуляция была произведена с помощью инструмента Intel Quartus Time Analyzer.



Также полученные модели были протестированы на ПЛИС семейства Intel Cyclon V и Aria 10, с подачей на них реальных сигналов из внешней среды, с оценкой временем обработки сигналов на ПВС.

