Федеральное государственное бюджетное учреждение науки Институт проблем проектирования в микроэлектронике Российской академии наук

Методы формирования и верификации библиотек стандартных элементов в составе маршрута проектирования ИС на базе ПЛИС отечественного производства

Хватов Василий Михайлович

Лялинская Ольга Владимировна

Гарбулина Татьяна Владимировна

Содержание

- Актуальность
- Цель работы
- Описание используемого маршрута проектирования
- Место представленных форматов библиотек в этом маршруте
- Способы формирования библиотек
- Описание методов их верификации
- Результаты и выводы

Актуальность

- На российском рынке стало появляться все больше ПЛИС отечественных производителей, с различными архитектурами;
- Для повышения скорости разработки ИС и их моделирования на ПЛИС производитель должен поставляться библиотеки стандартных элементов;
- Разработка и верификация библиотек стандартных элементов – обширная трудоёмкая задача.

Цель работы

- Представить типы библиотек используемые при проектировании устройств на базе отечественных ПЛИС;
- Описать процесс разработки и методы верификации каждого из рассмотренных форматов библиотек применительно к маршруту проектирования на ПЛИС с любой архитектурой;
- Разработка ПО для верификации преобразования Spice библиотеки в библиотеку на языке Verilog;

Описание разработанного маршрута проектирования

- Маршрут проектирования ИС на базе ПЛИС включает в себя ряд обязательных этапов (рис. 1):
- 1) разработка RTL-описания;
- 2) верификация RTL-описания;
- 3) проведение логического синтеза в САПР фирмы Synopsys – Design Compiler, в САПР фирмы Cadence – RTL Compiler, Genius или в ПО с открытой лицензией Yosys;
- 4) декомпозиция;
- 5) планировка / размещение на ПЛИС;
- 6) трассировка / конфигурирование элементов маршрутизации;
- 7) выполнение функциональной и временной верификации спроектированной схемы на базе ПЛИС.

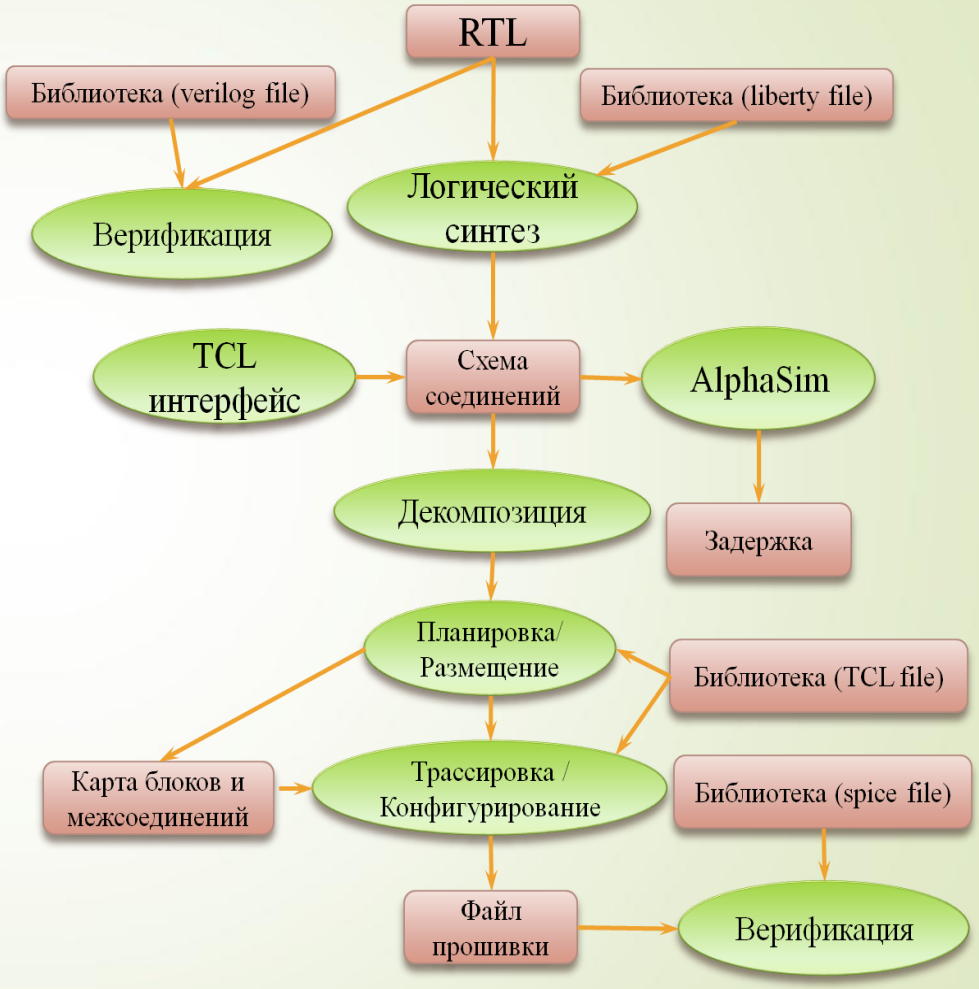

Рис. 1 Маршрут проектирования использованием лингвистических средств на базе языка TCL

Библиотека на языке Tcl

```
xc_lib_cell al_and2 LE_4p0 {
    a=A_Data
    b=C_Reset
    y=ELLR=LR
} {
     c_C=0
     c_CH=0
     c_CTr=0
     c_Rst=0
     c_Tr=0
     c_nA=1
     c_nB=1
     c_nC=0
     c_nQ=0
     c_nRst=1
     cn_A_g=0
     cn_B_g=1
     cn_CL=1
     cn_C_g=0
     cn_RstX=0
     nc_Tr=1
     nc_nA=0
     nc_nB=0
     nc_nQ=1
} {
    +f "y= a & b"
}
```

```
xc_lib_cell al_nor2ft LE_4p0 {
    a=A_Data
    b=C_Reset
    y=ELLR=LR
} {
     c_C=0
     c_CH=1
     c_CTr=0
     c_Rst=1
     c_Tr=0
     c_nA=1
     c_nB=1
     c_nC=0
     c_nQ=0
     c_nRst=0
     cn_A_g=0
     cn_B_g=1
     cn_CL=0
     cn_C_g=0
     cn_RstX=0
     nc_Tr=1
     nc_nA=0
     nc_nB=0
     nc_nQ=1
} {
    +f "y= !(!a | b)"
}
```

```
xc_lib_cell al_dff LE_4p0  {
    d=A_Data
    clk=B_Clk
    q=ELLR=LR
} {
          c_C=0
          c_CH=0
          c_CTr=1
          c_Rst=0
          c_Tr=1
          c_nA=0
          c_nB=0
          c_nC=0
          c_nQ=1
          c_nRst=0
          cn_A_g=0
          cn_B_g=0
          cn_CL=0
          cn_C_g=1
          cn_RstX=1
          nc_Tr=0
          nc_nA=1
          nc_nB=1
          nc_nQ=0
} {
    +f "q= !/clk & q | /clk & d"
}
```

Рис. 2 Примеры библиотечных элементов из Tcl-библиотеки

Библиотека на языке Spice

```
.subckt xc:al_and2 a b y
*
x_al_and2
+ a
+ gnd!
+ b
+ y
+ y<LR>
+ gnd!
+ gnd!
+ gnd!
+ gnd!
+ gnd!
+ vddh!
+ vddh!
+ gnd!
+ gnd!
+ vddh!
+ gnd!
+ vddh!
+ vddh!
+ gnd!
+ gnd!
+ vddh!
+ gnd!
+ gnd!
+ vddh!
.ends
```

```
.subckt xc:al_nor2ft a b y
*
x_al_nor2ft
+ a
+ gnd!
+ b
+ y
+ y<LR>
+ gnd!
+ vddh!
+ gnd!
+ vddh!
+ gnd!
+ vddh!
+ vddh!
+ gnd!
+ gnd!
+ gnd!
+ gnd!
+ vddh!
+ gnd!
+ gnd!
+ gnd!
+ vddh!
+ gnd!
+ gnd!
+ vddh!
+ LE_4p0
.ends
```

```
.subckt xc:al_dff d clk q
*
x_al_dff
+ d
+ clk
+ gnd!
+ q
+ q<LR>
+ gnd!
+ gnd!
+ vddh!
+ gnd!
+ vddh!
+ gnd!
+ gnd!
+ gnd!
+ vddh!
+ gnd!
+ gnd!
+ gnd!
+ gnd!
+ gnd!
+ vddh!
+ vddh!
+ gnd!
+ vddh!
+ vddh!
+ gnd!
+ LE_4p0
.ends
```

Рис. 3 Примеры библиотечных элементов из Spice-библиотеки

Характеризация

Рис. 4 Процесс характеризации с помощью AlphaSim

Библиотека формата lib

```
cell ( al_and2 ) {
    area : 2.0;
    cell_footprint : al;
    pin ( y ) {
    direction : output;
    function : "a&b";
    max_capacitance : 0.2;
    timing () {
        related_pin : "a";
        timing_sense : positive_unate;
    cell_rise {
        values( ... ); }
    rise_transition {
        values( ... ); }
    cell_fall {
        values( ... ); }
    fall_transition ( ) {
        values( ... ); }
    }
    timing () {
        related_pin : "b";…
```

```
cell ( al_nor2ft ) {
    area : 2.0;
    cell_footprint : al;
    pin ( y ) {
    direction : output;
    function : "!(!a|b)";
    max_capacitance : 0.2;
    timing () {
        related_pin : "b";
        timing_sense : negative_unate;
    cell_rise {
        values( ... ); }
    rise_transition {
        values( ... ); }
    cell_fall {
        values( ... ); }
    fall_transition ( ) {
        values( ... ); }
    }
    timing () {
        related_pin : "a";…
```

```
cell ( al_dff ) {
    area : 2.0;
    cell_footprint : al;
        ff(Iq,Iq1) {
            next_state : "d";
            clocked_on : "clk";
            clear      : "clr";
        }
    pin ( q ) {
    direction : output;
    function : "Iq";
    max_capacitance : 0.2;
    timing () {
        related_pin : "clk";
        timing_type : rising_edge;
    cell_fall {
        values( ... ); }
    fall_transition ( ) {
        values( ... ); }
    }
    cell_rise {
        values( ... ); }
    rise_transition {
        values( ... ); }

    timing () {
        related_pin : "clr";
        timing_sense : negative_unate;
        timing_type : clear;
```

Рис. 5 Примеры библиотечных элементов из lib-библиотеки

Библиотека на языке Verilog

```
module al_and2 (y, a, b);
    output y;
    input a, b;

    // Function
    and (y, a, b);

    // Timing
    specify
        (a => y) = 0.1;
        (b => y) = 0.1;
    endspecify
endmodule
```

```
module al_nor2ft (y, a, b);
    output y;
    input a, b;

    // Function
    wire a__bar, int_fwire_0;

    not (a__bar, a);
    or (int_fwire_0, a__bar, b);
    not (y, int_fwire_0);

    // Timing
    specify
        (a => y) = 0;
        (b => y) = 0;
    endspecify
endmodule
```

```
module al_dff (q, d, clk);
    output q;
    input d, clk;
    reg notifier;
    wire delayed_d, delayed_clk;

    // Function
    wire int_fwire_Iq, xcr_0;

    XD_dff_err (xcr_0, delayed_clk, delayed_d);
    XD_dff (int_fwire_Iq, notifier, delayed_clk, delayed_d, xcr_0);
    buf (q, int_fwire_Iq);

    // Timing
    specify
        (posedge clk => (q+:d)) = 0;
        $setuphold (posedge clk, posedge d, 0, 0, notifier,,, delayed_clk, delayed_d);
        $setuphold (posedge clk, negedge d, 0, 0, notifier,,, delayed_clk, delayed_d);
        $width (posedge clk, 0, 0, notifier);
        $width (negedge clk, 0, 0, notifier);
    endspecify
endmodule
```

Рис. 6 Примеры библиотечных элементов из Verilog-библиотеки

Верификация

- 1) **Tcl** – верифицируется разработчиком;
- 2) **Tcl » Spice** – автоматическая генерация, основа для .lib и .v библиотек и .v библиотек;
- 3) **Spice » Lib** – верификация с помощью двух последующих методов;
- 4) **Lib » Verilog** – верификация Cadence Conformal;
- 5) **Spice » Verilog** – верификация разработанным ПО языке Tcl

Выводы и результаты

- В данной работе были представлены методы формирования библиотек на языке Tcl, Spice, Verilog и общая структура характеризации библиотечных элементов и получения liberty файла.
- Были описаны существующие способы верификации Tcl, Verilog и .lib библиотек, а также разработаны программные средства верификации библиотек формата Spice и Verilog.