TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN - ĐIỆN TỬ

BÁO CÁO BÀI TẬP LỚN VLSI

Đề tài:

THIẾT KẾ BỘ ĐẾM TĂNG GIẢM

Giáo viên hướng dẫn: **TS. Nguyễn Vũ Thắng**

Sinh viên thực hiện: Nguyễn Văn Lưu – 20192993

Bùi Thức Hậu – 20192830

Nguyễn Quang Đông - 20192760

Hà Nội, 07 -2023

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin ngày nay, vi mạch đã trở thành trái tim của hầu hết các thiết bị điện tử mà chúng ta sử dụng hàng ngày. Từ điện thoại thông minh, máy tính cá nhân, đến thiết bị gia đình thông minh và các hệ thống tự động hoá công nghiệp, vi mạch đóng vai trò quan trọng trong việc xử lý thông tin và điều khiển các chức năng cơ bản và phức tạp.

Từ những kiến thức học được, nhóm sinh viên chúng em quyết định thực hiện đề tài " **Thiết kế bộ đếm tăng giảm**" do công ty Dolphin đưa ra nhằm mục đích học tập trả nghiệm các quy trình phát triển một sản phẩm về vi mạch. Trong quá trình thực hiện đề tài nhóm nghiên cứu đã nỗ lực, nghiêm túc thực hiện cùng sự hướng dẫn của giáo viên hướng dẫn, song chắc chắn không tránh khỏi những hạn chế. Nhóm nghiên cứu rất mong nhận được những ý kiến đóng góp của thầy cô và các bạn sinh viên. Qua quá trình nghiên cứu cũng xin cảm ơn Thầy Nguyễn Vũ Thắng và các anh bên công ty Dolphin đã tận tình giúp nhóm nghiên cứu hoàn thành sản phẩm.

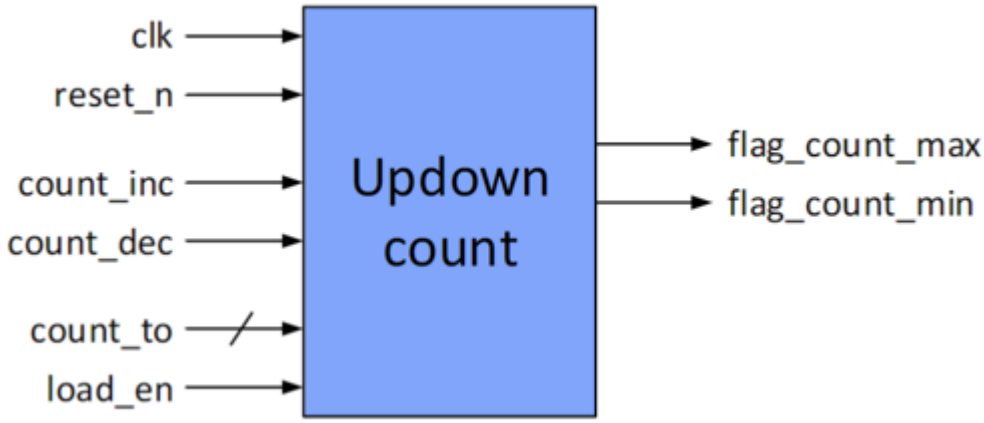
MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

CHƯƠNG 1. THIẾT KẾ KIẾN TRÚC

1.1 Sơ đồ khối



Hình 1. Sơ đồ khối Counter Module

1.2 Chức năng

Biến counter thể hiện giá trị của bộ đếm. Khi nhận được tín hiệu reset_n = 0 thì counter được gán bằng 0, khi tín hiệu reset_n = 1 thì thực hiện chức năng tăng giảm của bộ đếm.

Khi nhận được tín hiệu load_en = 1 thực hiện load count_to vào bộ, khi tín hiệu load_en = 0 thì vẫn thực hiện chức năng tăng giảm bộ đếm. Giá trị của counter sẽ phụ thuộc hai tín hiệu count_inc và count_dec như sau:

count_inc	count_dec	counter
0	0	Không đổi
0	1	-1
1	0	+1
1	1	Không đổi

Bảng 1. Bảng chức năng tăng giảm

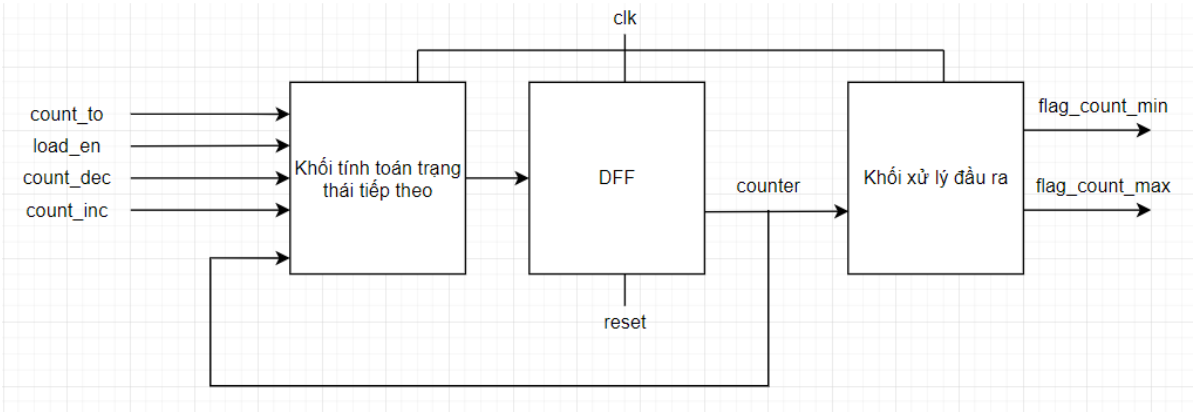
Khi counter đạt giá trị tối đa (count_to) thì tín hiệu đầu ra flag_count_max = 1 và tín hiệu đầu ra flag_count_min = 0, khi counter = 0 thì tín hiệu đầu ra flag_count_max = 0 và tín hiệu đầu ra flag_count_min = 1, khi 0 < counter < count_to thì tín hiệu đầu ra flag_count_max = 0 và tín hiệu đầu ra flag_count_min = 0.

1.3 Đầu vào đầu ra

Tên tín hiệu	Số bit	Input/Output	Mô tả
clk	1	Input	Clock đầu vào
reset_n	1	Input	Reset không đồng bộ, tích cực mức thấp
count_to	4	Input	Giá trị tối đa của bộ đếm
count_inc	1	Input	Điều khiển bộ đếm tăng
count_dec	1	Input	Điều khiển bộ đếm giảm
load_en	1	Input	Tải giá trị count_to vào bộ đếm
flag_count_max	1	Output	Cờ báo bộ đếm đã đạt giá tối đa (count_to)
flag_count_min	1	Output	Cờ báo bộ đếm đã đạt giá trị tối thiểu (0)

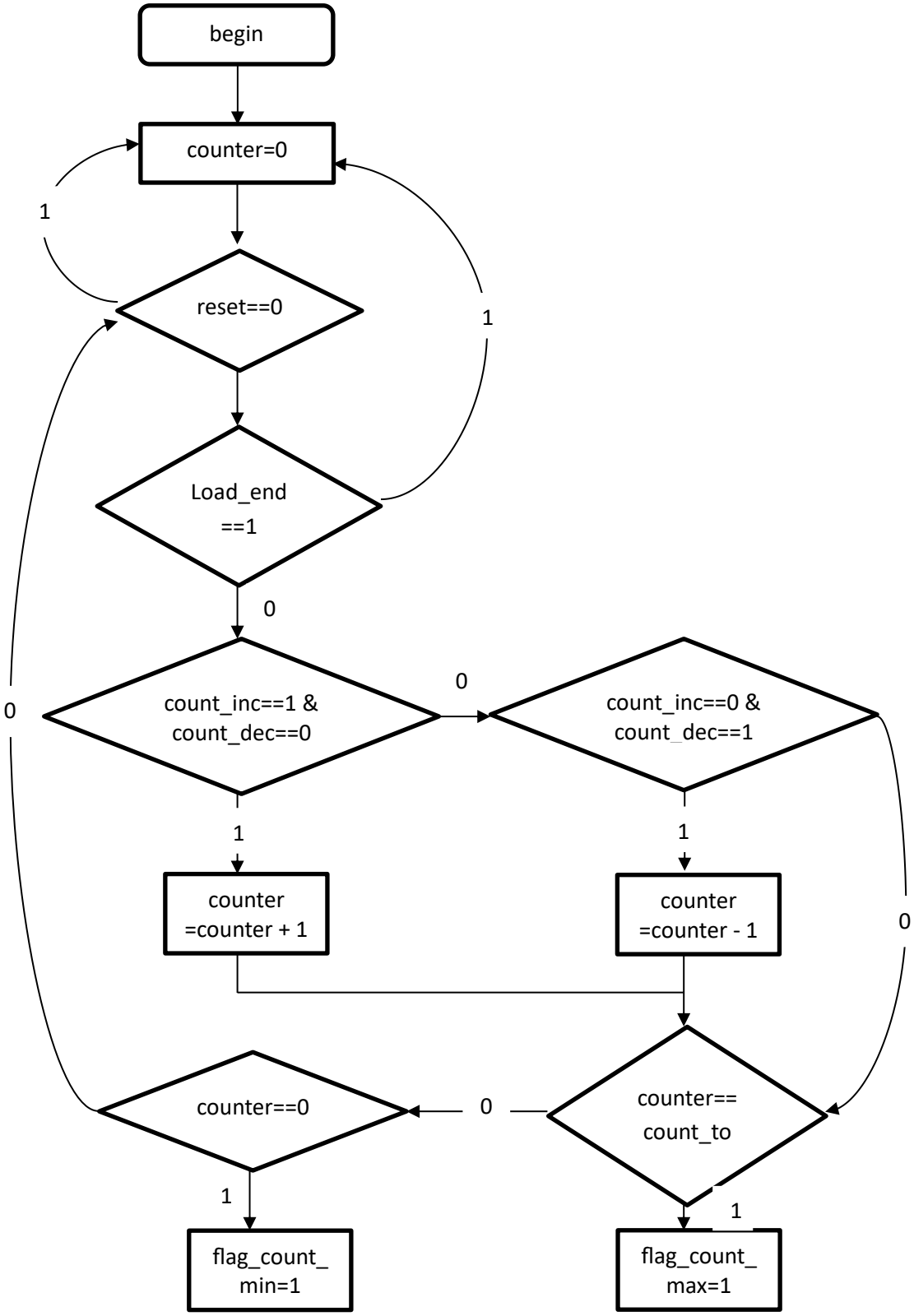
Bảng 2. Mô tả đầu vào đầu ra

1.4 Kiến trúc tổng quát



Hình 2. Kiến trúc cho thiết kế

1.5 Lưu đồ thuật toán



Hình 3: Lưu đồ thuật toán

CHƯƠNG 2. TRIỂN KHAI HDL

2.1 Thiết kế bằng Verilog

Sau khi thiết kế xong kiến trúc thì chúng em triển khai bằng ngôn ngữ mô tả phần cứng Verilog

```
module Counter(
  input wire clk,
  input wire reset_n,
  input wire [3:0] count_to,
  input wire count_inc,
  input wire count_dec,
  input wire load_en,
  output reg flag_count_max,
  output reg flag_count_min
);

  reg [3:0] counter;
  reg [3:0] count_max;

  always @(posedge clk or negedge reset_n) begin
    if (~reset_n) begin
      counter <= 4'b0000;
    end else begin
      if (load_en) begin
        counter <= 4'b0000;
        count_max <= count_to;
      end else begin
        if (count_inc & ~count_dec) begin
          counter <= (counter == count_max) ? counter : counter + 1;
        end else if (~count_inc & count_dec) begin
          counter <= (counter == 4'b0000) ? counter : counter - 1;
        end
      end
    end
  end

  always @(posedge clk) begin
    flag_count_max <= (counter == count_max);
    flag_count_min <= (counter == 4'b0000);
  end

endmodule
```

Hình 4: File triển khai bằng verilog

Để kiểm tra xem thiết kế đã đúng với yêu cầu hay chưa bằng cách viết 1 file test bench

```
module TestBench;

  reg clk;
  reg reset_n;
  reg [3:0] count_to;
  reg count_inc;
  reg count_dec;
  reg load_en;
  wire flag_count_max;
  wire flag_count_min;

  // Instantiate the Counter module
  Counter counter_inst (
    .clk(clk),
    .reset_n(reset_n),
    .count_to(count_to),
    .count_inc(count_inc),
    .count_dec(count_dec),
    .load_en(load_en),
    .flag_count_max(flag_count_max),
    .flag_count_min(flag_count_min)
  );

  // Clock generation
  always begin
    #5 clk = ~clk;
  end

  // Initialize inputs
  initial begin
    clk = 0;
    reset_n = 1;
    count_to = 4'b1111;
    count_inc = 0;
    count_dec = 0;
    load_en = 0;
```

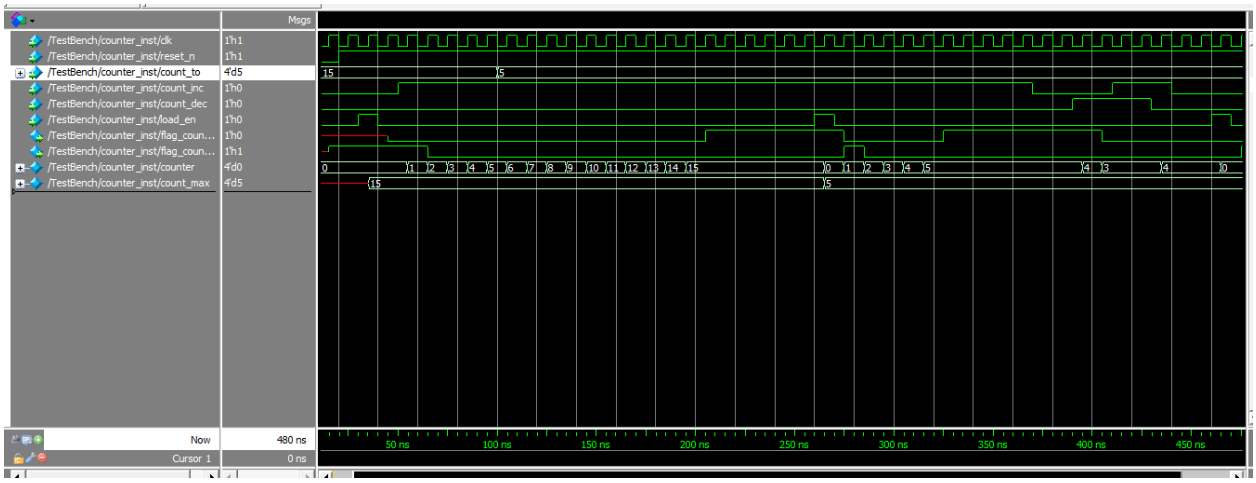
```
#10 reset_n = 0;
#10 reset_n = 1;
#10 load_en = 0;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
count_to = 4'b0101;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_inc = 1;
#10 load_en = 1;
#10 load_en = 0;
#10 count_inc = 1;
#10 count_inc = 1;
#10 count_dec = 0;
#20 count_inc = 0;
#20 count_dec = 1;
#20 count_inc = 1;
#20 count_dec = 0;
#10 count_inc = 0;
#10 count_dec = 0;

#10 load_en = 1;
#10 load_en = 0;

#10 $finish;
```

Hình 5: Triển khai testbench

2.2 Kết quả kiểm thử



Hình 6. Kết quả mô phỏng

Nhận xét: Kết quả đúng với thiết kế của kiến trúc

CHƯƠNG 3. TỔNG HỢP RTL

3.1 Tool tổng hợp Yosys

Yosys là phần mềm mã nguồn mở dùng để tổng hợp RTL, chuyển đổi verilog sang BLIF / EDIF/ BTOR / SMT-LIB / Verilog RTL đơn giản. Yosys hỗ trợ tổng hợp ánh xạ theo thư viện cell tiêu chuẩn.

Ở trong bài tập lớn này chúng em sử dụng thư viện cell của công ty Dolphin cung cấp:

dti_tm55gp_60_10t_stdcells_ff_1p1v_0c_rev1p0p1.lib

```
timing() {
  related_pin : "A";
  timing_type : combinational;
  timing_sense : positive_unate;
  cell_rise(dti_delay_template) {
    index_1 ("0.0009000, 0.0046720, 0.0116800, 0.0292000, 0.0730000, 0.1825000, 0.5475000");
    index_2 ("0.0004300, 0.0010630, 0.0026300, 0.0065040, 0.0160850, 0.0397790, 0.1193370");
    values ( \
      "0.0119390, 0.0134339, 0.0166316, 0.0237750, 0.0410019, 0.0824451, 0.2244571", \
      "0.0127926, 0.0142695, 0.0174509, 0.0245501, 0.0414878, 0.0831689, 0.2248862", \
      "0.0144452, 0.0158929, 0.0190647, 0.0262903, 0.0431495, 0.0847519, 0.2242238", \
      "0.0174503, 0.0189449, 0.0222580, 0.0294279, 0.0464567, 0.0881345, 0.2313683", \
      "0.0215623, 0.0232612, 0.0267894, 0.0342556, 0.0512546, 0.0931207, 0.2331681", \
      "0.0267686, 0.0287341, 0.0329401, 0.0412113, 0.0592891, 0.1015529, 0.2421171", \
      "0.0332034, 0.0357963, 0.0414275, 0.0527164, 0.0749382, 0.1216759, 0.2663710"  \
    );
  }
  cell_fall(dti_delay_template) {
    index_1 ("0.0009000, 0.0046720, 0.0116800, 0.0292000, 0.0730000, 0.1825000, 0.5475000");
    index_2 ("0.0004300, 0.0010630, 0.0026300, 0.0065040, 0.0160850, 0.0397790, 0.1193370");
    values ( \
      "0.0094611, 0.0105464, 0.0127621, 0.0176009, 0.0286369, 0.0564291, 0.1474684", \
      "0.0105221, 0.0116010, 0.0139200, 0.0187544, 0.0298785, 0.0573436, 0.1485466", \
      "0.0123744, 0.0134742, 0.0157985, 0.0208022, 0.0318744, 0.0590254, 0.1502292", \
      "0.0156984, 0.0169571, 0.0194304, 0.0245245, 0.0356854, 0.0629024, 0.1540300", \
      "0.0200977, 0.0216369, 0.0245414, 0.0303289, 0.0419904, 0.0690532, 0.1597470", \
      "0.0257519, 0.0275327, 0.0314333, 0.0384167, 0.0520813, 0.0807989, 0.1717646", \
      "0.0338497, 0.0363079, 0.0411815, 0.0515898, 0.0705006, 0.1056133, 0.2052939"  \
    );
  }
  rise_transition(dti_delay_template) {
    index_1 ("0.0009000, 0.0046720, 0.0116800, 0.0292000, 0.0730000, 0.1825000, 0.5475000");
    index_2 ("0.0004300, 0.0010630, 0.0026300, 0.0065040, 0.0160850, 0.0397790, 0.1193370");
    values ( \
      "0.0047069, 0.0062909, 0.0097577, 0.0191819, 0.0441522, 0.1031123, 0.3207429", \
      "0.0047342, 0.0064577, 0.0100391, 0.0192585, 0.0434165, 0.1043951, 0.3175222", \
      "0.0049050, 0.0063987, 0.0100210, 0.0193872, 0.0434857, 0.1059897, 0.3161122", \
      "0.0055829, 0.0069503, 0.0107801, 0.0194996, 0.0447622, 0.1041203, 0.3143669", \
      "0.0075361, 0.0089327, 0.0123509, 0.0206314, 0.0436701, 0.1054155, 0.3096676", \
      "0.0119358, 0.0136425, 0.0165733, 0.0247288, 0.0470123, 0.1051507, 0.3081017", \
      "0.0251169, 0.0267894, 0.0309163, 0.0386739, 0.0596907, 0.1169239, 0.3120494"  \
```

Hình 7: Thư viện cell tiêu chuẩn

3.2 Tiến hành tổng hợp

Đầu tiên export file verilog chính là bản thiết kế HDL của nhóm.

Sau đó link với thư viện cell tiêu chuẩn.

Sau khi tổng hợp xong thì ghi ra file “netlist” chính là file verilog chứa các con cell cơ bản như FlipFlop, AND, OR, XOR, NOT, NOR.

Cuối cùng là xuất ra file report chứa thông tin về tên cell, số lượng cell cần dùng để thuận tiện cho việc vẽ layout.

```
module Counter(clk, reset_n, count_to, count_inc, count_dec, load_en, flag_count_max, flag_count_min);
  (* src = "counter.v:32.3-35.6" *)
  wire _000_;
  (* src = "counter.v:32.3-35.6" *)
  wire _001_;
  wire _002_;
  wire _003_;
  wire _004_;
  wire _005_;
  wire _006_;
  wire _007_;
  wire _008_;
  wire _009_;
  wire _010_;
  wire _011_;
  wire _012_;
  wire _013_;
  wire _014_;
  wire _015_;
  wire _016_;
  wire _017_;
  wire _018_;
  wire _019_;
  wire _020_;
  wire _021_;
  wire _022_;
  wire _023_;
  wire _024_;
  wire _025_;
  wire _026_;
  wire _027_;
  wire _028_;
  wire _029_;
  wire _030_;
  wire _031_;
  wire _032_;
  wire _033_;
  wire _034_;
  wire _035_;
  wire _036_;
  wire _037_;
  wire _038_;
  wire _039_;
  wire _040_;
  wire _041_;
  wire _042_;
  wire _043_;
  wire _044_;
  wire _045_;
  wire _046_;
  wire _047_;
  wire _048_;
  wire _049_;
  wire _050_;
  wire _051_;
  wire _052_;
  wire _053_;
  wire _054_;
  wire _055_;
```

```
(* src = "counter.v:2.14-2.17" *)
input clk;
wire clk;
(* src = "counter.v:6.14-6.23" *)
input count_dec;
wire count_dec;
(* src = "counter.v:5.14-5.23" *)
input count_inc;
wire count_inc;
(* src = "counter.v:13.13-13.22" *)
wire [3:0] count_max;
(* src = "counter.v:4.20-4.28" *)
input [3:0] count_to;
wire [3:0] count_to;
(* src = "counter.v:12.13-12.20" *)
wire [3:0] counter;
(* src = "counter.v:8.14-8.28" *)
output flag_count_max;
wire flag_count_max;
(* src = "counter.v:9.14-9.28" *)
output flag_count_min;
wire flag_count_min;
(* src = "counter.v:7.14-7.21" *)
input load_en;
wire load_en;
(* src = "counter.v:3.14-3.21" *)
input reset_n;
wire reset_n;
dti_55g_10t_invx1 _056_ (
  .A(count_dec),
  .Z(_010_)
);
dti_55g_10t_invx1 _057_ (
  .A(counter[0]),
  .Z(_011_)
);
dti_55g_10t_invx1 _058_ (
  .A(counter[2]),
  .Z(_012_)
);
dti_55g_10t_xor2x1 _059_ (
  .A(count_max[1]),
  .B(counter[1]),
  .Z(_013_)
);
dti_55g_10t_xor2x1 _060_ (
  .A(count_max[0]),
  .B(counter[0]),
  .Z(_014_)
);
dti_55g_10t_xor2x1 _061_ (
  .A(count_max[3]),
  .B(counter[3]),
  .Z(_015_)
);
dti_55g_10t_xor2x1 _062_ (
  .A(count_max[2]),
  .B(counter[2]),
  .Z(_016_)
);
```

```
dti_55g_10t_or3x1 _063_ (
  .A(_013_),
  .B(_014_),
  .C(_016_),
  .Z(_017_)
);
dti_55g_10t_nor2x1 _064_ (
  .A(_015_),
  .B(_017_),
  .Z(_000_)
);
dti_55g_10t_or2x1 _065_ (
  .A(counter[0]),
  .B(counter[1]),
  .Z(_018_)
);
dti_55g_10t_nor3x1 _066_ (
  .A(counter[2]),
  .B(counter[3]),
  .C(_018_),
  .Z(_001_)
);
dti_55g_10t_and2x1 _067_ (
  .A(load_en),
  .B(reset_n),
  .Z(_019_)
);
dti_55g_10t_nand2x1 _068_ (
  .A(load_en),
  .B(reset_n),
  .Z(_020_)
);
dti_55g_10t_nand2x1 _069_ (
  .A(count_max[0]),
  .B(_020_),
  .Z(_021_)
);
dti_55g_10t_nand2x1 _070_ (
  .A(count_to[0]),
  .B(_019_),
  .Z(_022_)
);
dti_55g_10t_nand2x1 _071_ (
  .A(_021_),
  .B(_022_),
  .Z(_002_)
);
dti_55g_10t_nand2x1 _072_ (
  .A(count_max[1]),
  .B(_020_),
  .Z(_023_)
);
dti_55g_10t_nand2x1 _073_ (
  .A(count_to[1]),
  .B(_019_),
  .Z(_024_)
);
dti_55g_10t_nand2x1 _074_ (
  .A(_023_),
  .B(_024_),
  .Z(_003_)
);
dti_55g_10t_nand2x1 _075_ (
  .A(count_max[2]),
  .B(_020_),
  .Z(_025_)
);
dti_55g_10t_nand2x1 _076_ (
  .A(count_to[2]),
  .B(_019_),
  .Z(_026_)
);
dti_55g_10t_nand2x1 _077_ (
  .A(_025_),
  .B(_026_),
  .Z(_004_)
);
```

```
);
dti_55g_10t_nand2x1 _078_ (
  .A(count_max[3]),
  .B(_020_),
  .Z(_027_)
);
dti_55g_10t_nand2x1 _079_ (
  .A(count_to[3]),
  .B(_019_),
  .Z(_028_)
);
dti_55g_10t_nand2x1 _080_ (
  .A(_027_),
  .B(_028_),
  .Z(_005_)
);
dti_55g_10t_and2x1 _081_ (
  .A(count_inc),
  .B(_010_),
  .Z(_029_)
);
dti_55g_10t_nand2x1 _082_ (
  .A(count_inc),
  .B(_010_),
  .Z(_030_)
);
dti_55g_10t_nor2x1 _083_ (
  .A(_000_),
  .B(_030_),
  .Z(_031_)
);
dti_55g_10t_nor3x1 _084_ (
  .A(count_inc),
  .B(_010_),
  .C(_001_),
  .Z(_032_)
);
dti_55g_10t_nor3x1 _085_ (
  .A(load_en),
  .B(_031_),
  .C(_032_),
  .Z(_033_)
);
dti_55g_10t_nand2x1 _086_ (
  .A(counter[0]),
  .B(_033_),
  .Z(_034_)
);
dti_55g_10t_nor2x1 _087_ (
  .A(load_en),
  .B(_033_),
  .Z(_035_)
);
dti_55g_10t_nand2x1 _088_ (
  .A(_011_),
  .B(_035_),
  .Z(_036_)
);
dti_55g_10t_nand2x1 _089_ (
  .A(_034_),
  .B(_036_),
  .Z(_006_)
);
dti_55g_10t_nand2x1 _090_ (
  .A(counter[1]),
  .B(_033_),
  .Z(_037_)
);
dti_55g_10t_nand2x1 _091_ (
  .A(counter[1]),
  .B(_030_),
  .Z(_038_)
);
dti_55g_10t_xor2x1 _092_ (
  .A(counter[1]),
  .B(_029_),
  .Z(_039_)
);
```

```
);
dti_55g_10t_nand2x1 _093_ (
  .A(_011_),
  .B(_039_),
  .Z(_040_)
);
dti_55g_10t_or2x1 _094_ (
  .A(_011_),
  .B(_039_),
  .Z(_041_)
);
dti_55g_10t_nand3x1 _095_ (
  .A(_035_),
  .B(_040_),
  .C(_041_),
  .Z(_042_)
);
dti_55g_10t_nand2x1 _096_ (
  .A(_037_),
  .B(_042_),
  .Z(_007_)
);
dti_55g_10t_nand2x1 _097_ (
  .A(counter[2]),
  .B(_033_),
  .Z(_043_)
);
dti_55g_10t_nand2x1 _098_ (
  .A(_038_),
  .B(_041_),
  .Z(_044_)
);
dti_55g_10t_nor2x1 _099_ (
  .A(_012_),
  .B(_030_),
  .Z(_045_)
);
dti_55g_10t_nor2x1 _100_ (
  .A(counter[2]),
  .B(_029_),
  .Z(_046_)
);
dti_55g_10t_or2x1 _101_ (
  .A(_045_),
  .B(_046_),
  .Z(_047_)
);
dti_55g_10t_xor2x1 _102_ (
  .A(_044_),
  .B(_047_),
  .Z(_048_)
);
dti_55g_10t_nand2x1 _103_ (
  .A(_035_),
  .B(_048_),
  .Z(_049_)
);
dti_55g_10t_nand2x1 _104_ (
  .A(_043_),
  .B(_049_),
  .Z(_008_)
);
dti_55g_10t_nor3x1 _105_ (
  .A(_012_),
  .B(_030_),
  .C(_041_),
  .Z(_050_)
);
dti_55g_10t_and3x1 _106_ (
  .A(_038_),
  .B(_041_),
  .C(_046_),
  .Z(_051_)
);
```

```
);
dti_55g_10t_nor2x1 _107_ (
  .A(_050_),
  .B(_051_),
  .Z(_052_)
);
dti_55g_10t_nor2x1 _108_ (
  .A(_033_),
  .B(_052_),
  .Z(_053_)
);
dti_55g_10t_and2x1 _109_ (
  .A(counter[3]),
  .B(_053_),
  .Z(_054_)
);
dti_55g_10t_nor2x1 _110_ (
  .A(counter[3]),
  .B(_053_),
  .Z(_055_)
);
dti_55g_10t_nor3x1 _111_ (
  .A(load_en),
  .B(_054_),
  .C(_055_),
  .Z(_009_)
);
(* src = "counter.v:15.3-30.6" *)
dti_55g_10t_ffqbcka01x1 _112_ (
  .CK(clk),
  .D(_002_),
  .Q(count_max[0]),
  .RN(1'h1)
);
(* src = "counter.v:15.3-30.6" *)
dti_55g_10t_ffqbcka01x1 _113_ (
  .CK(clk),
  .D(_003_),
  .Q(count_max[1]),
  .RN(1'h1)
);
(* src = "counter.v:15.3-30.6" *)
dti_55g_10t_ffqbcka01x1 _114_ (
  .CK(clk),
  .D(_004_),
  .Q(count_max[2]),
  .RN(1'h1)
);
(* src = "counter.v:15.3-30.6" *)
dti_55g_10t_ffqbcka01x1 _115_ (
  .CK(clk),
  .D(_005_),
  .Q(count_max[3]),
  .RN(1'h1)
);
(* src = "counter.v:32.3-35.6" *)
dti_55g_10t_ffqbcka01x1 _116_ (
  .CK(clk),
  .D(_000_),
  .Q(flag_count_max),
  .RN(1'h1)
);
(* src = "counter.v:32.3-35.6" *)
dti_55g_10t_ffqbcka01x1 _117_ (
  .CK(clk),
  .D(_001_),
  .Q(flag_count_min),
  .RN(1'h1)
);
(* src = "counter.v:15.3-30.6" *)
dti_55g_10t_ffqbcka01x1 _118_ (
  .CK(clk),
  .D(_006_),
  .Q(counter[0]),
  .RN(reset_n)
);
```

```
  );
  (* src = "counter.v:15.3-30.6" *)
  dti_55g_10t_ffqbcka01x1 _119_ (
    .CK(clk),
    .D(_007_),
    .Q(counter[1]),
    .RN(reset_n)
  );
  (* src = "counter.v:15.3-30.6" *)
  dti_55g_10t_ffqbcka01x1 _120_ (
    .CK(clk),
    .D(_008_),
    .Q(counter[2]),
    .RN(reset_n)
  );
  (* src = "counter.v:15.3-30.6" *)
  dti_55g_10t_ffqbcka01x1 _121_ (
    .CK(clk),
    .D(_009_),
    .Q(counter[3]),
    .RN(reset_n)
  );
endmodule
```

Hình 8: file netlist sau khi tổng hợp

```
=== Counter ===

   Number of wires:                 66
   Number of wire bits:             75
   Number of public wires:          10
   Number of public wire bits:      19
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                 66
     dti_55g_10t_and2x1              3
     dti_55g_10t_and3x1              1
     dti_55g_10t_ffqbcka01x1        10
     dti_55g_10t_invx1               3
     dti_55g_10t_nand2x1            25
     dti_55g_10t_nand3x1             1
     dti_55g_10t_nor2x1              8
     dti_55g_10t_nor3x1              5
     dti_55g_10t_or2x1               3
     dti_55g_10t_or3x1               1
     dti_55g_10t_xor2x1              6
```

Hình 9: Thông tin về cell và số lượng cell

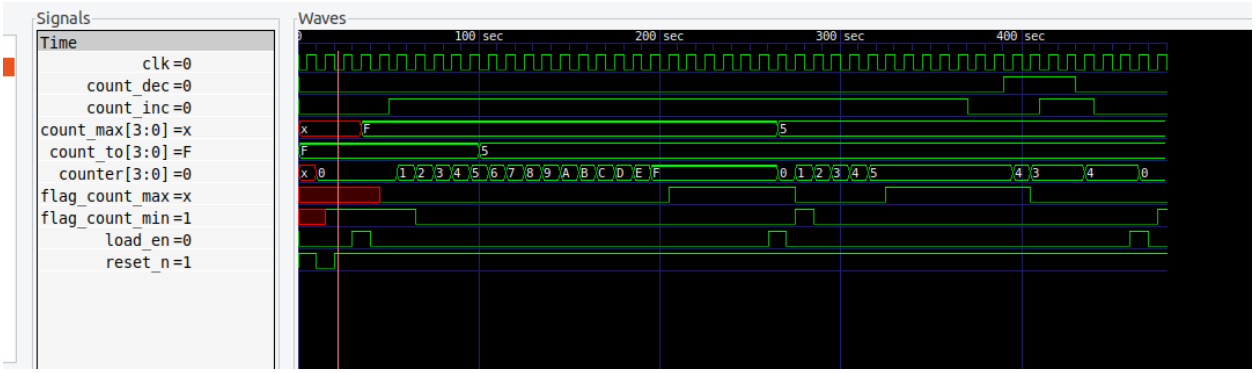
3.3 Kiểm tra lại kết quả tổng hợp

Sau khi tổng hợp xong cần kiểm tra lại file RTL tổng hợp đã đúng với file thiết kế verilog ban đầu.

Sử dụng file verilog dti_tm55gp_60_10t_stdcells_rev1p0p1.v là file triển khai của thư viện cell tiêu chuẩn được công ty cung cấp và file TestBench ban đầu trong bước triển khai HDL.

```
cu_cai@cu-cai-Nitro-AN515-54:~/VLSI/yosys/kiem_tra$ ls
compile  COUNTER.VCD  dti_tm55gp_60_10t_stdcells_rev1p0p1.v  netlist.v  TestBench.v
cu_cai@cu-cai-Nitro-AN515-54:~/VLSI/yosys/kiem_tra$
```

Và đây là kết quả :



Hình 10: Kết quả wave của bước kiểm tra

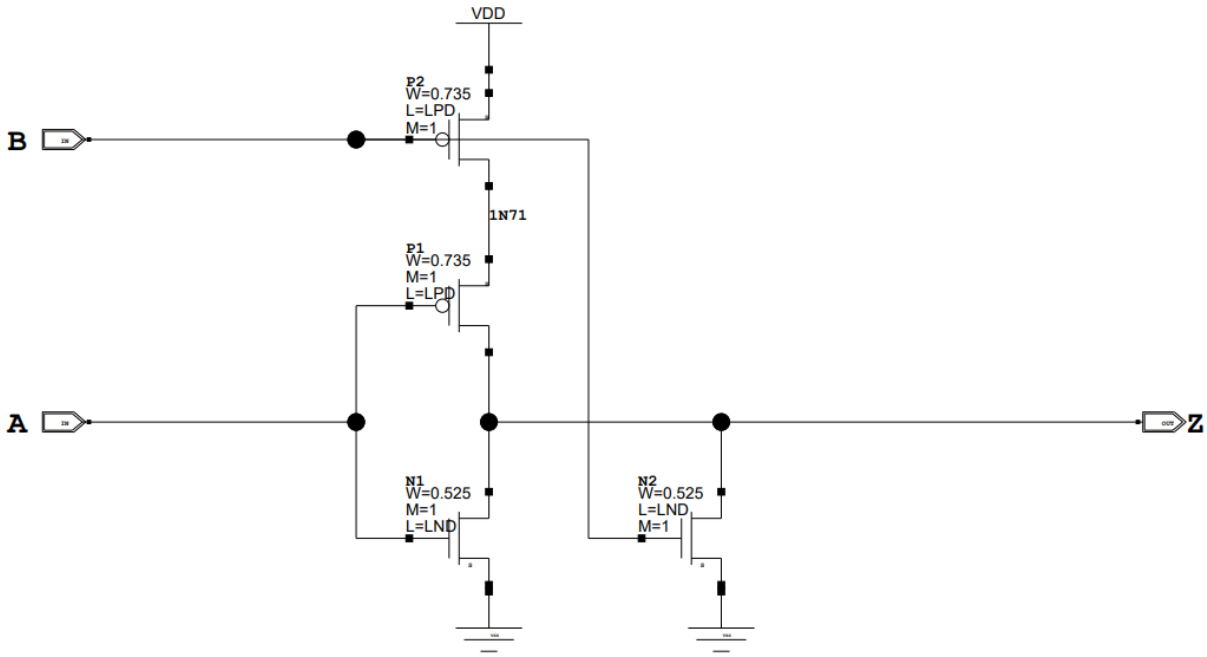
Nhận xét: Kết quả song của hai triển khai là như nhau, vì vậy sẽ tiến hành vẽ layout cho thiết kế .

CHƯƠNG 4. LAYOUT

4.1 Một số cổng logic cơ bản

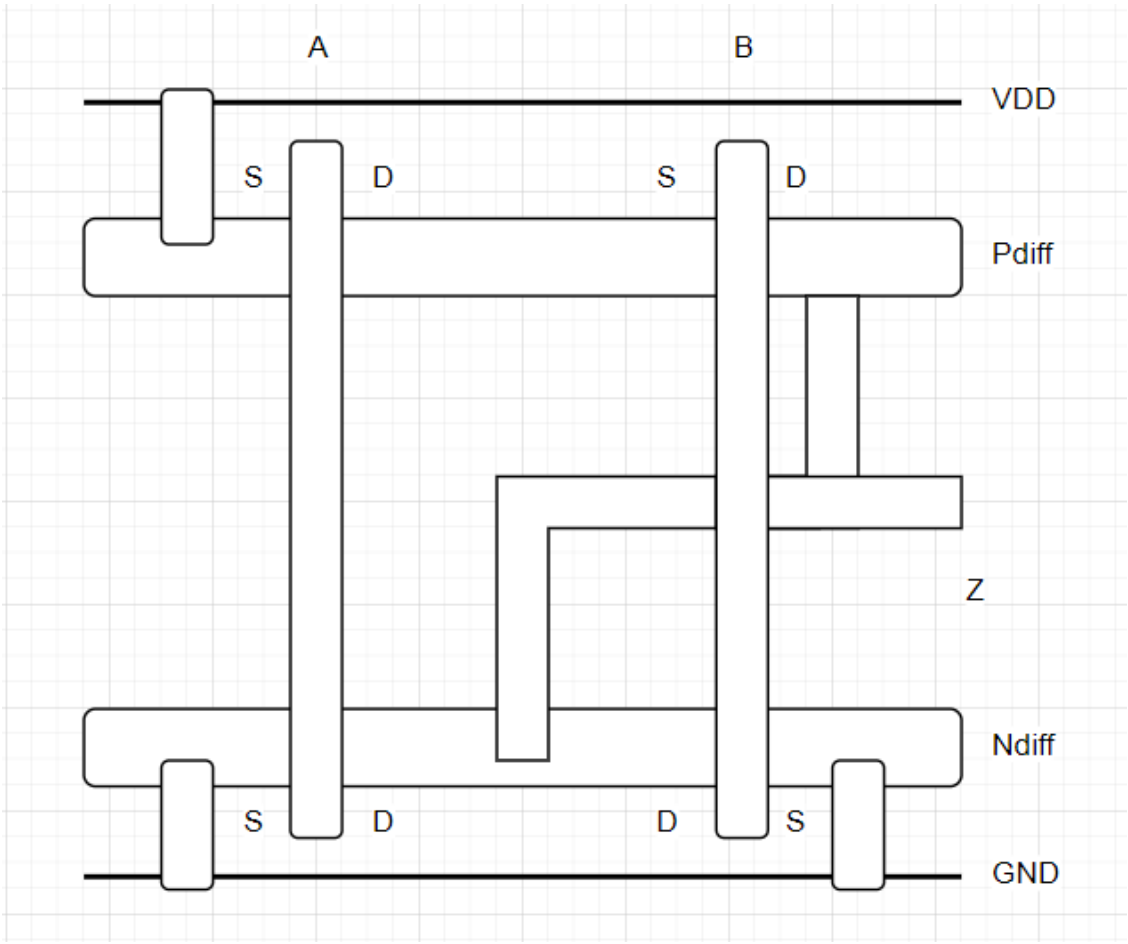
4.1.1 NOR 2x1

4.1.1.1 Mạch nguyên lý



Hình 11. Mạch nguyên lý cổng NOR 2x1

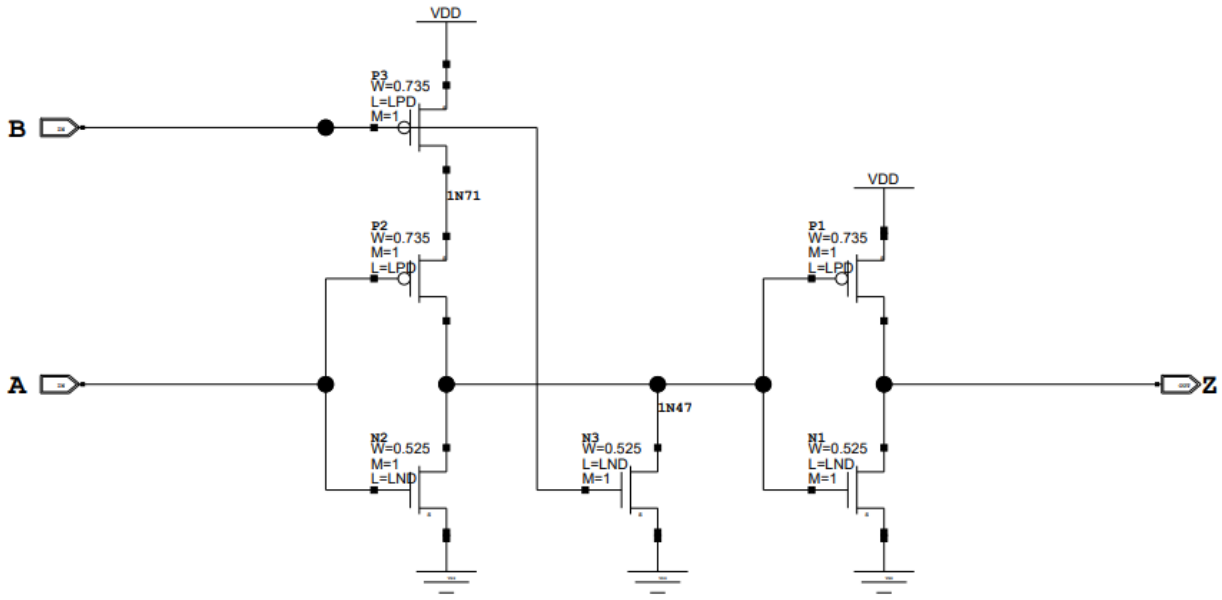
4.1.1.2 Stick diagram



Hình 12. Stick diagram NOR 2x1

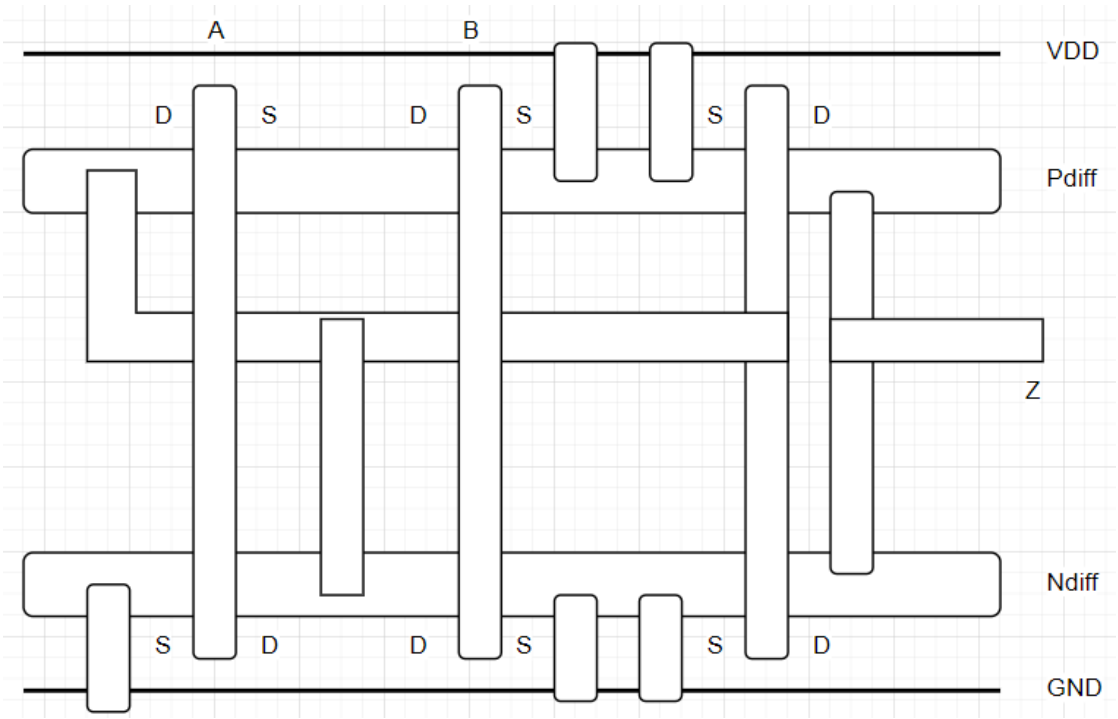
4.1.2 OR 2x1

4.1.2.1 Mạch nguyên lý



Hình 13. Mạch nguyên lý OR 2x1

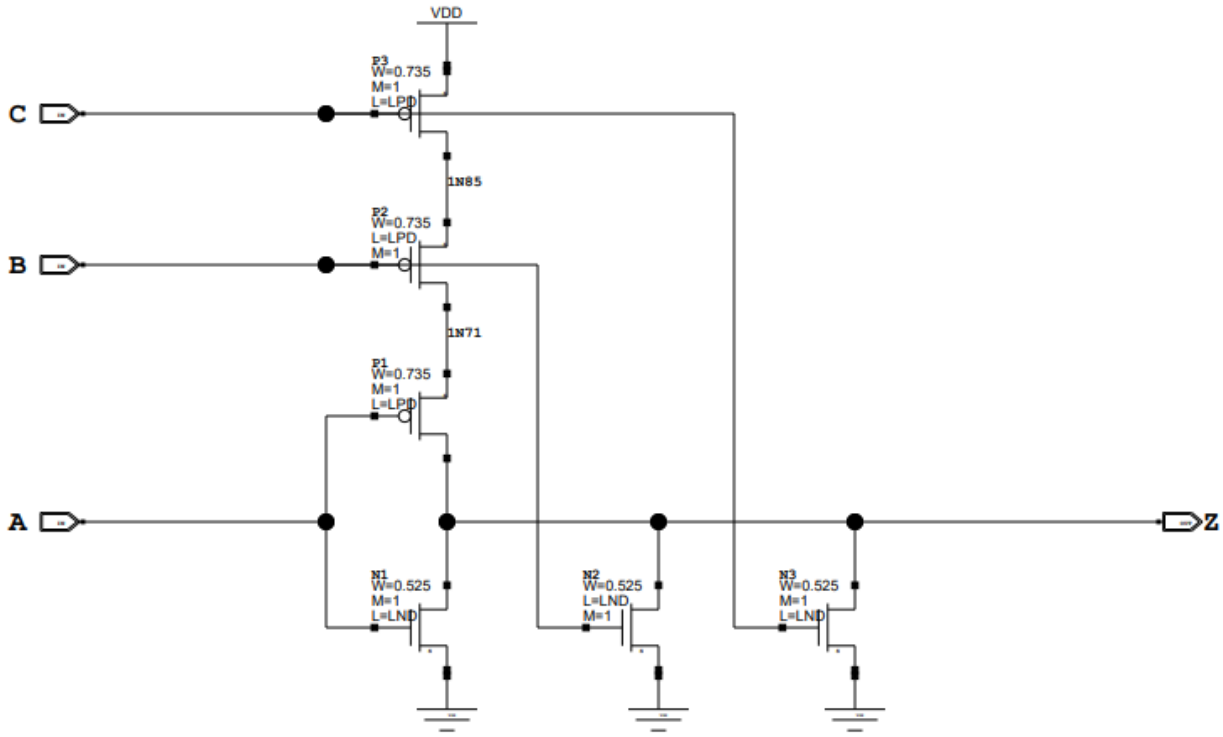
4.1.2.2 Stick diagram



Hình 14. Stick diagram OR 2x1

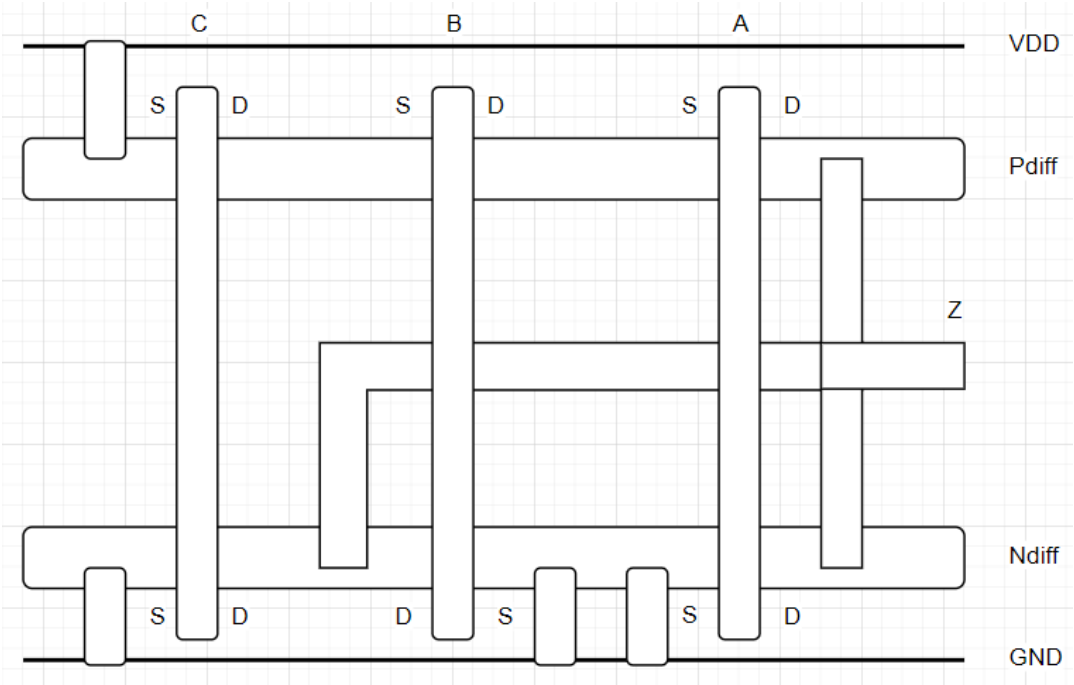
4.1.3 NOR 3x1

4.1.3.1 Mạch nguyên lý



Hình 15. Mạch nguyên lý NOR 3x1

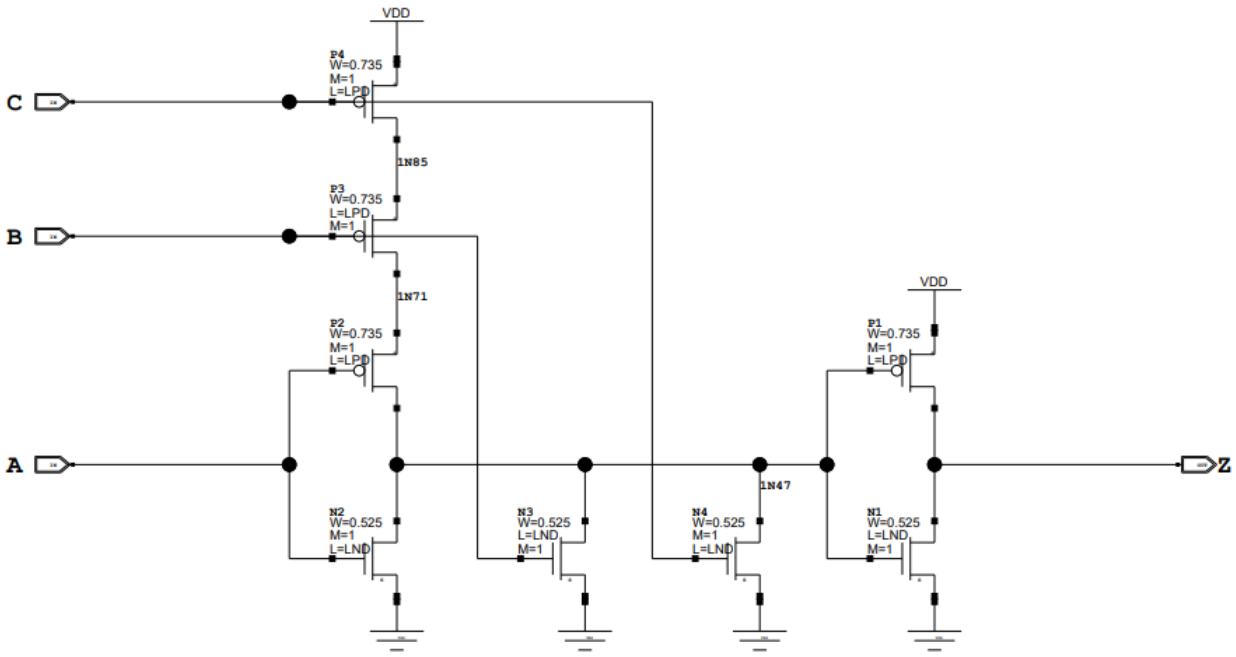
4.1.3.2 Stick diagram



Hình 16. Stick diagram NOR 3x1

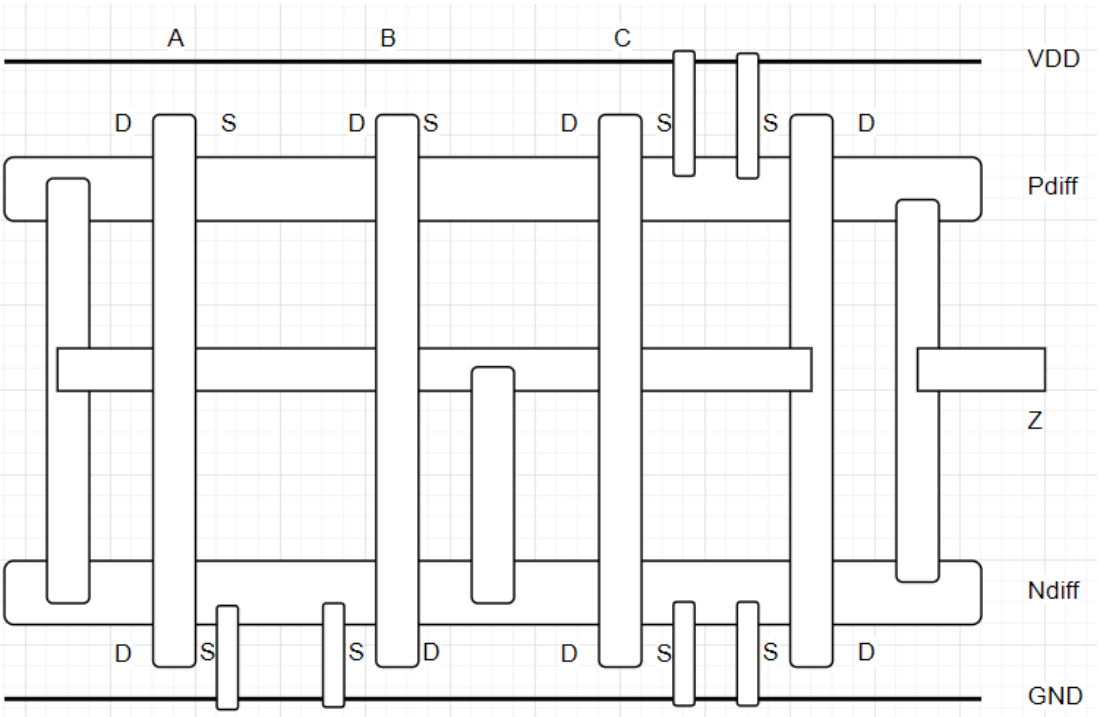
4.1.4 OR 3x1

4.1.4.1 Mạch nguyên lý



Hình 17. Mạch nguyên lý OR 3x1

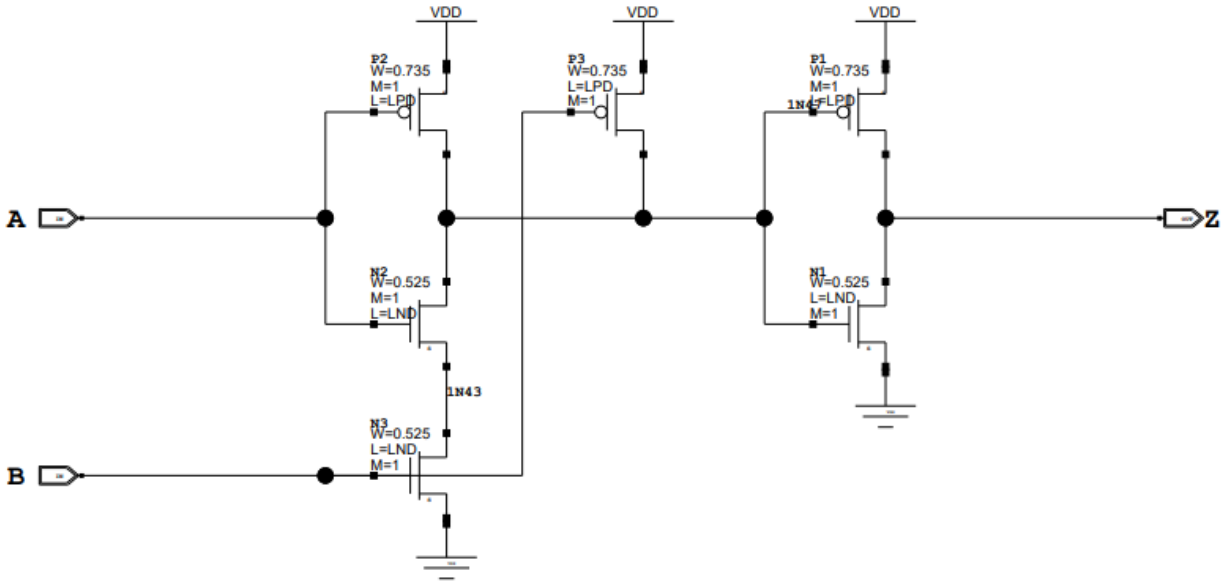
4.1.4.2 Stick diagram



Hình 18. Stick diagram OR 3x1

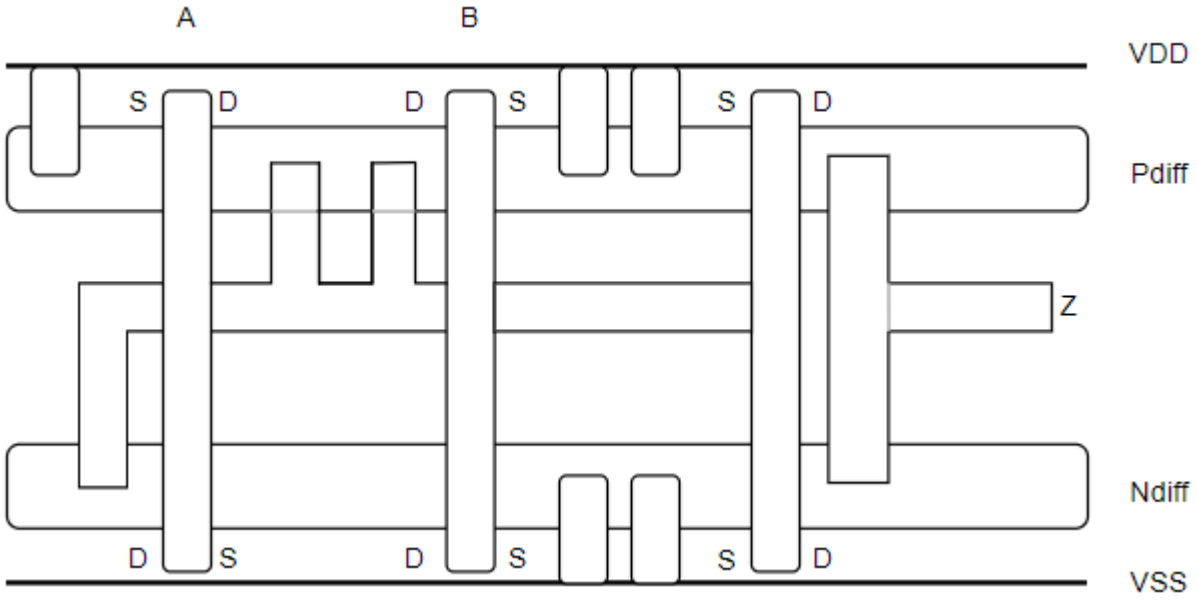
4.1.5 AND 2x2

4.1.5.1 Mạch nguyên lý



Hình 19. Mạch nguyên lý AND 2x1

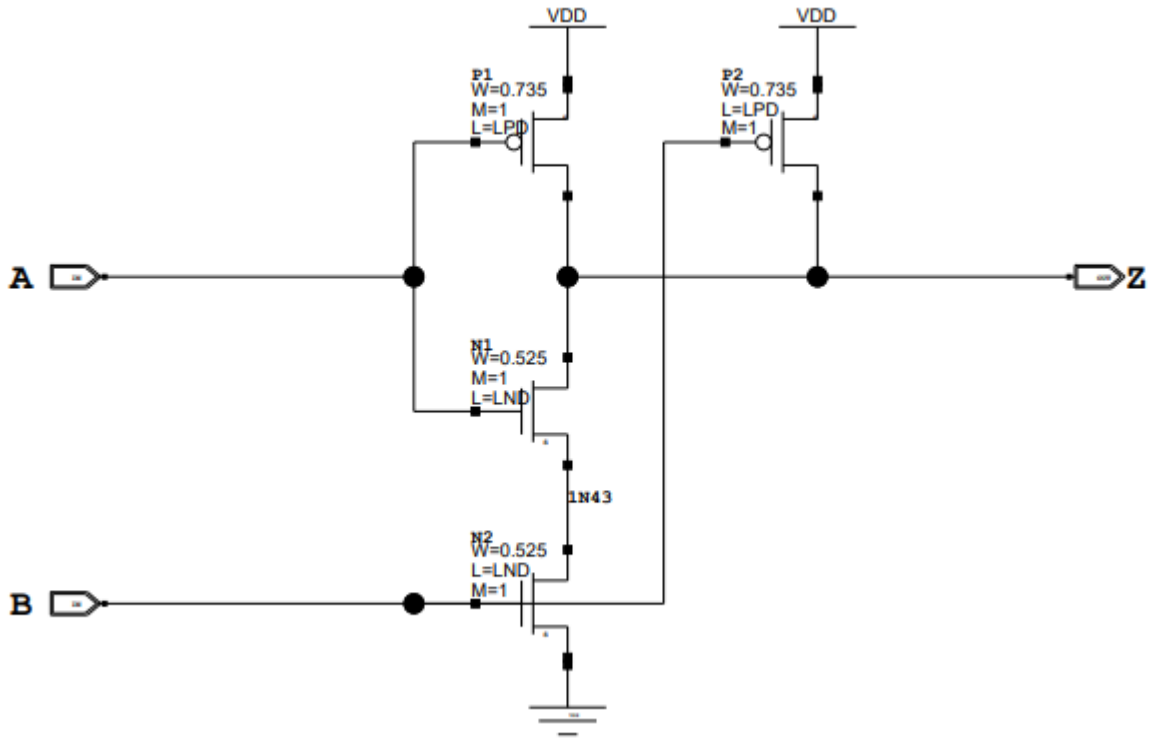
4.1.5.2 Stick diagram



Hình 20. Stick diagram AND 2x1

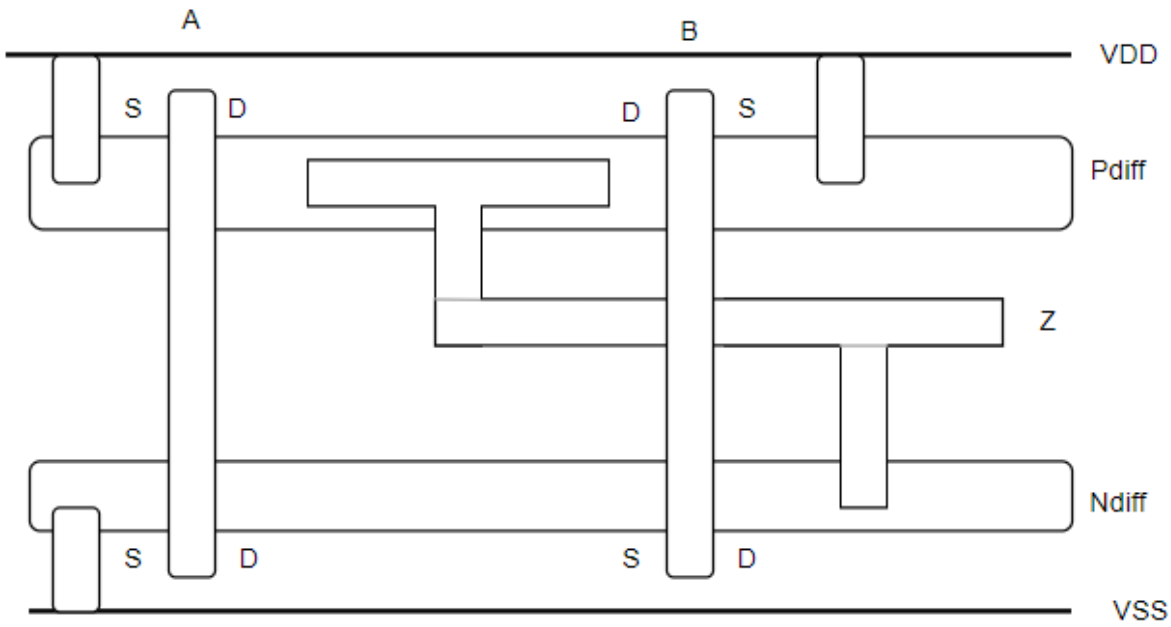
4.1.6 NAND 2x2

4.1.6.1 Mạch nguyên lý



Hình 21. Mạch nguyên lý NAND 2x1

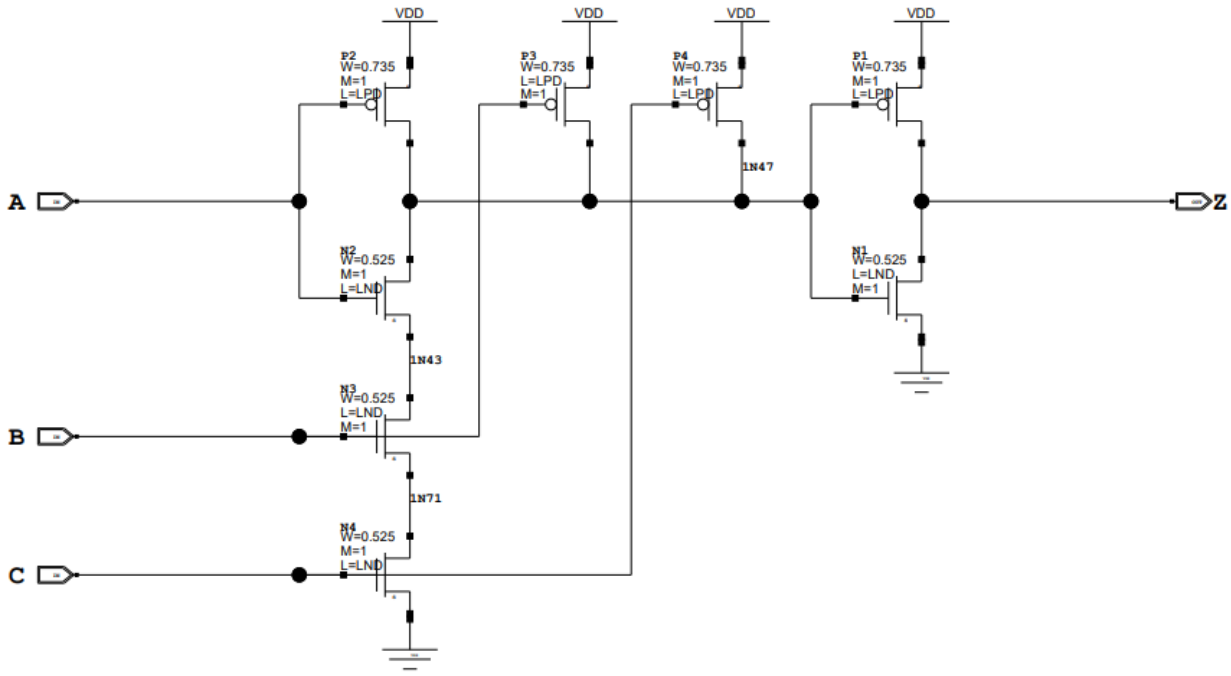
4.1.6.2 Stick diagram



Hình 22. Stick diagram NAND 2x1

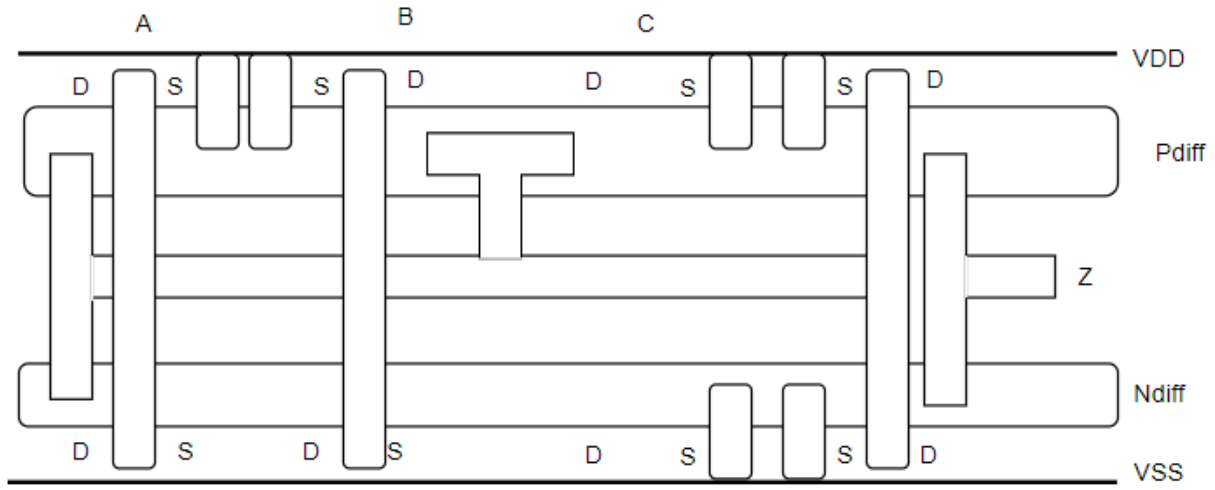
4.1.7 AND 3x1

4.1.7.1 Mạch nguyên lý



Hình 23. Mạch nguyên lý AND 3x1

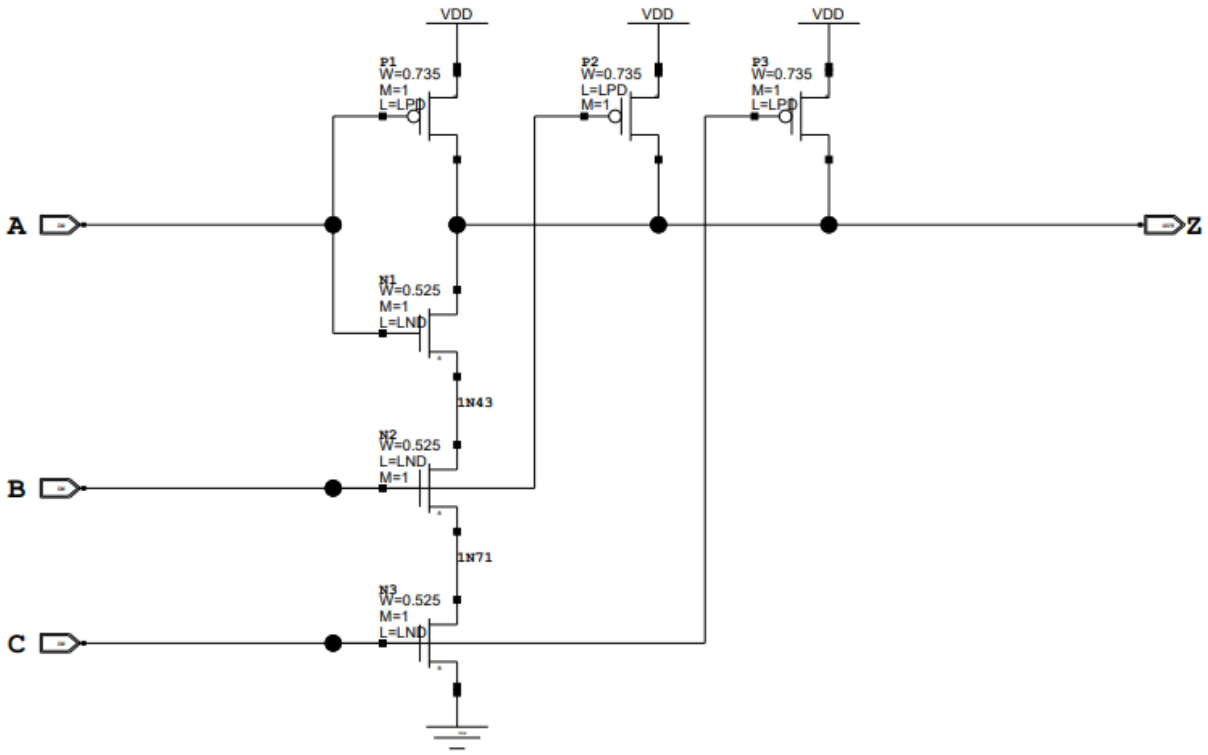
4.1.7.2 Stick diagram



Hình 24. Stick diagram AND 3x1

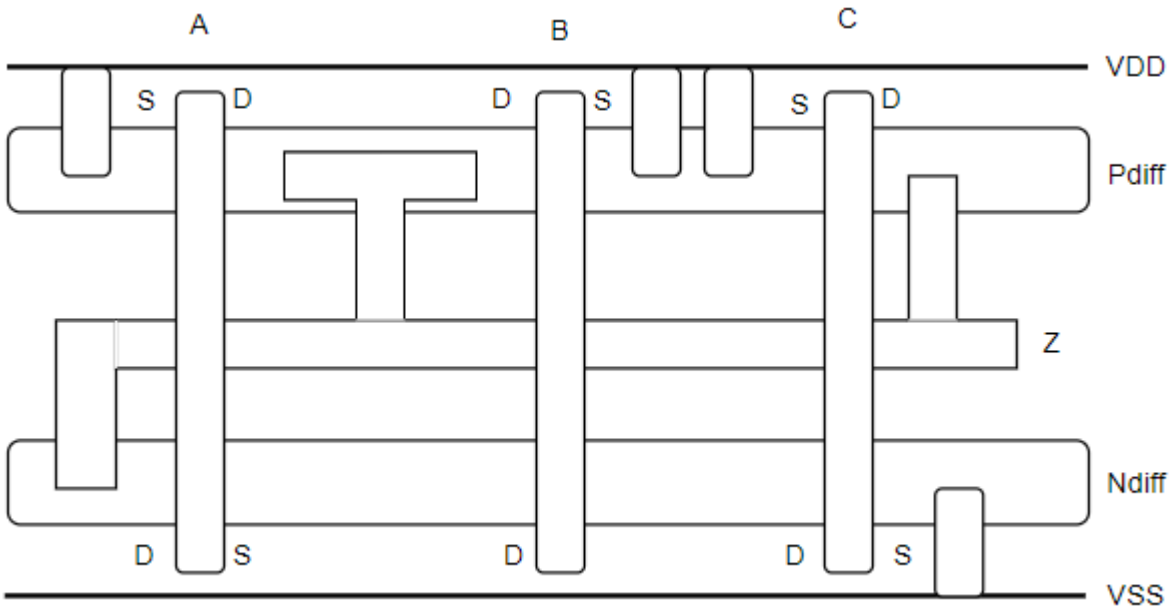
4.1.8 NAND 3x1

4.1.8.1 Mạch nguyễn lý



Hình 25. Mạch nguyên lý NAND 3x1

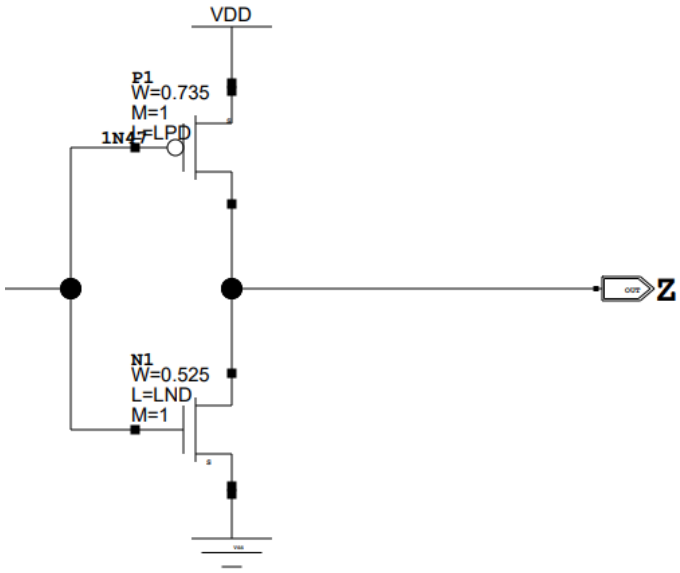
4.1.8.2 Stick diagram



Hình 26. Stick diagram NAND 3x1

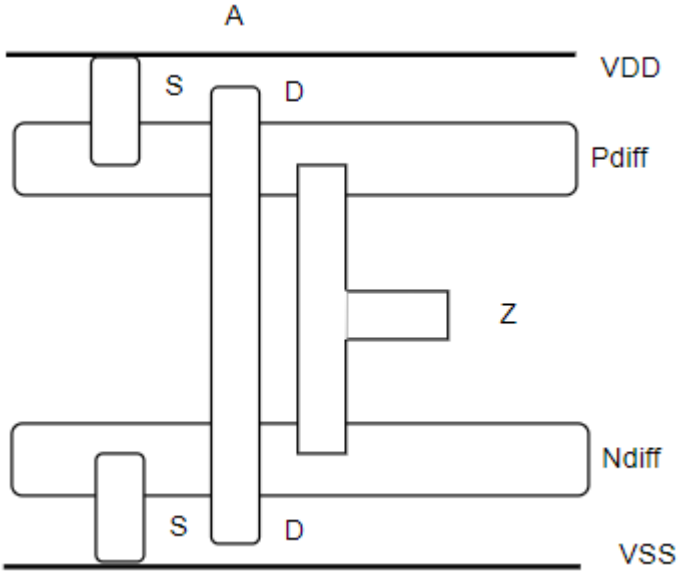
4.1.9 NOT 1x1

4.1.9.1 Mạch nguyên lý



Hình 27. Mạch nguyên lý NOT 1x1

4.1.9.2 Stick diagram



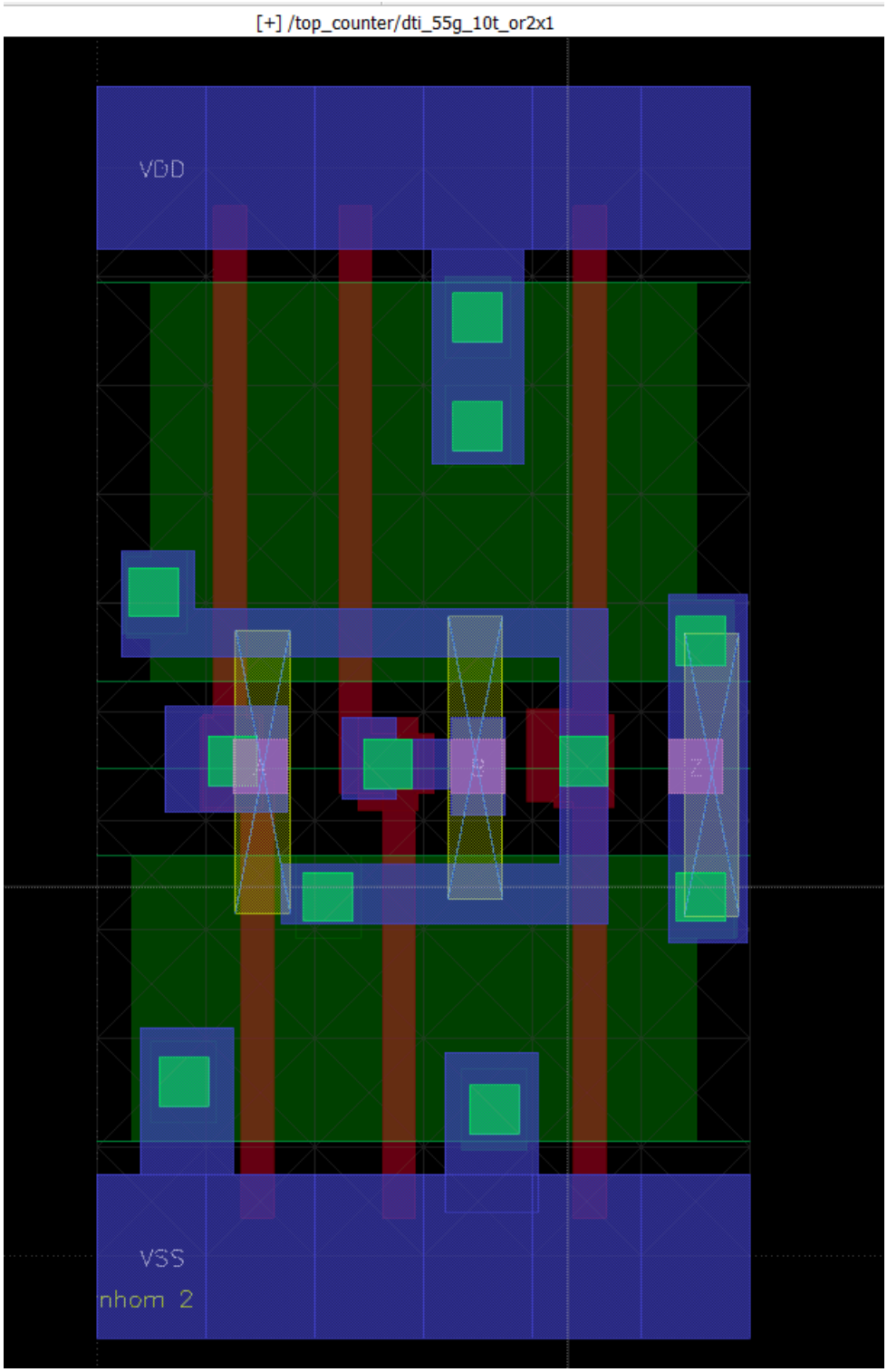
Hình 28. Stick diagram NOT 1x1

4.2 Tiến hành vẽ Layout

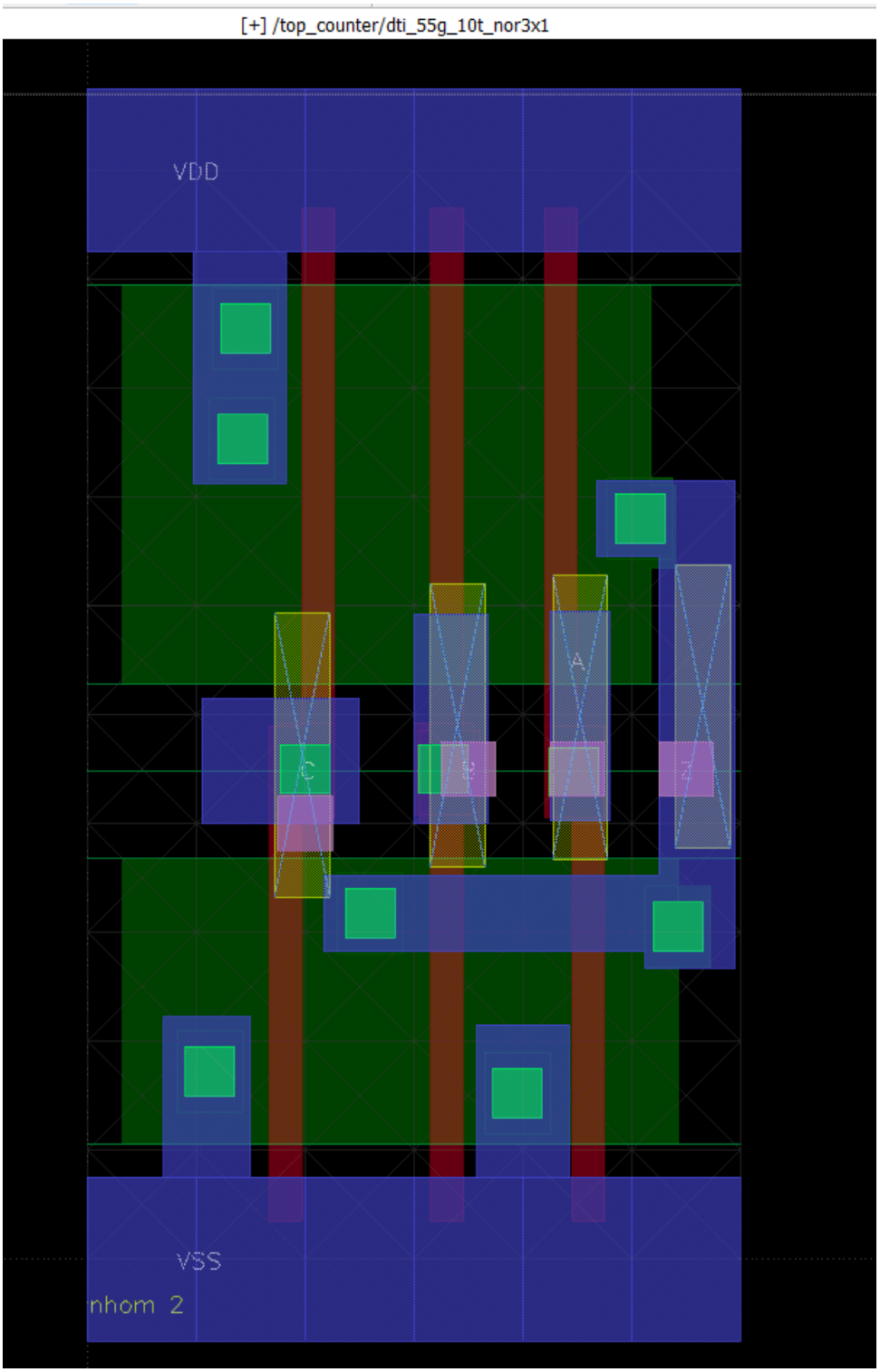
Layout được vẽ trên phần mềm chuyên dụng vẽ layout do công ty Dolphin cung cấp

Công nghệ vẽ layout là công nghệ 10 track 55mm phù hợp với sự yêu cầu công ty

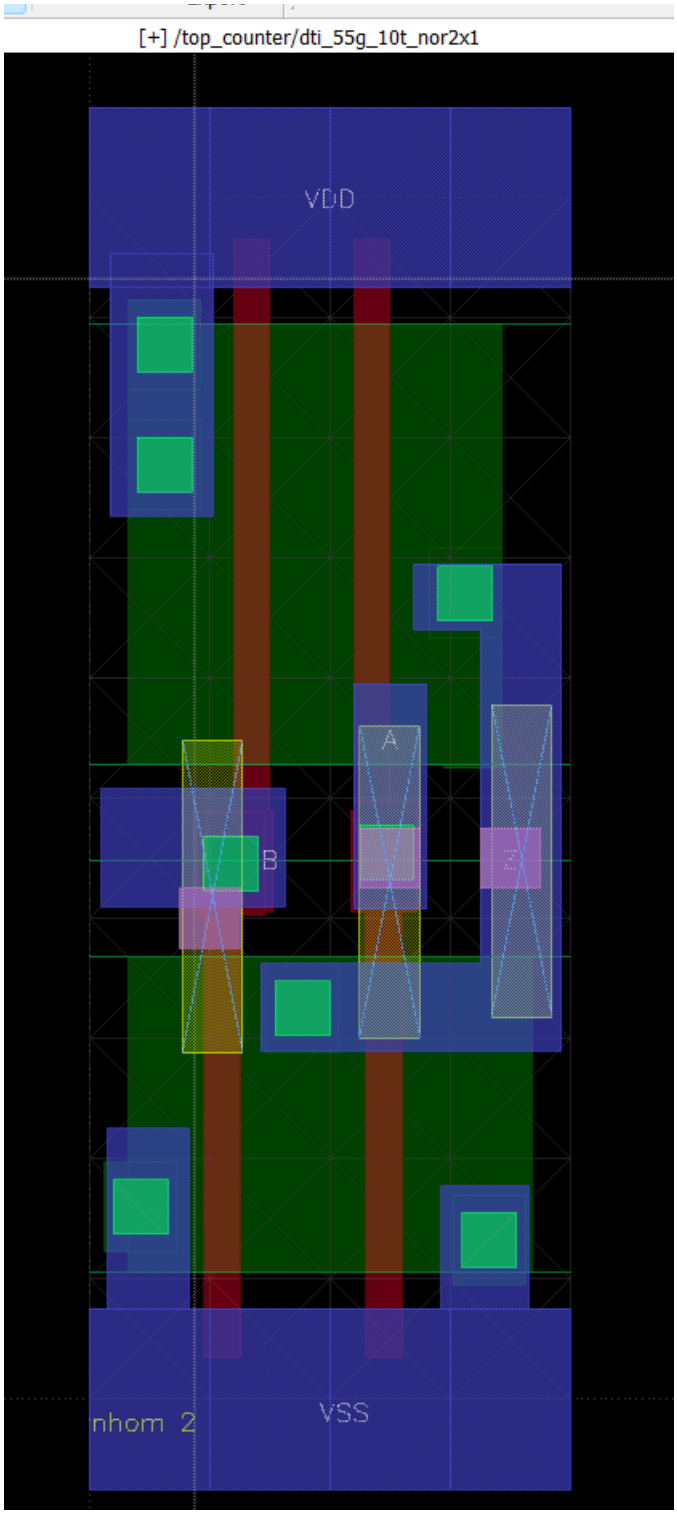
Kết quả Layout:



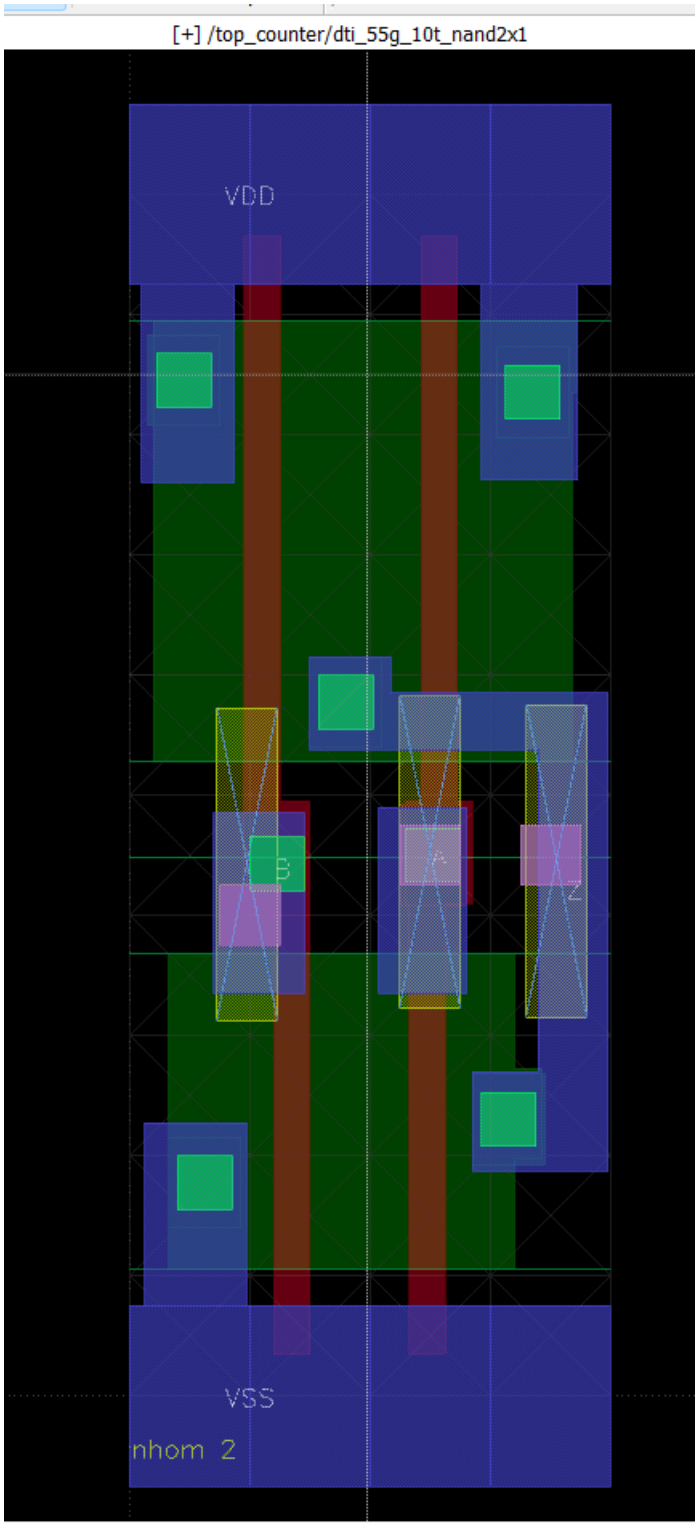
Hình 29. Hình kết quả 1



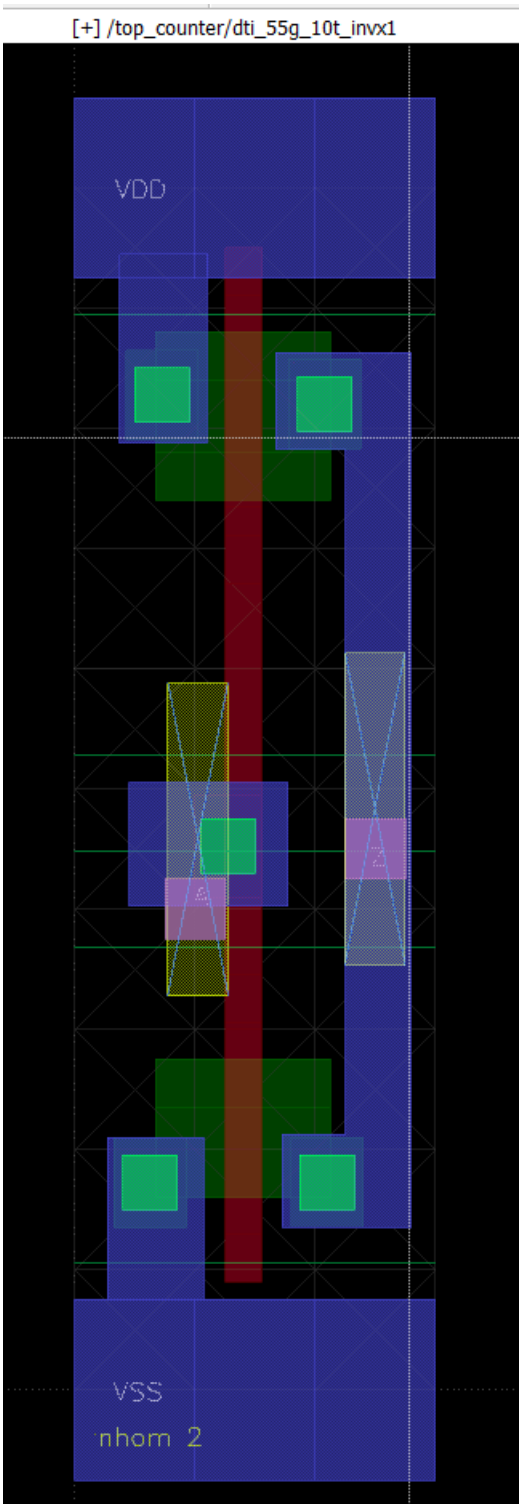
Hình 30. Hình kết quả 2



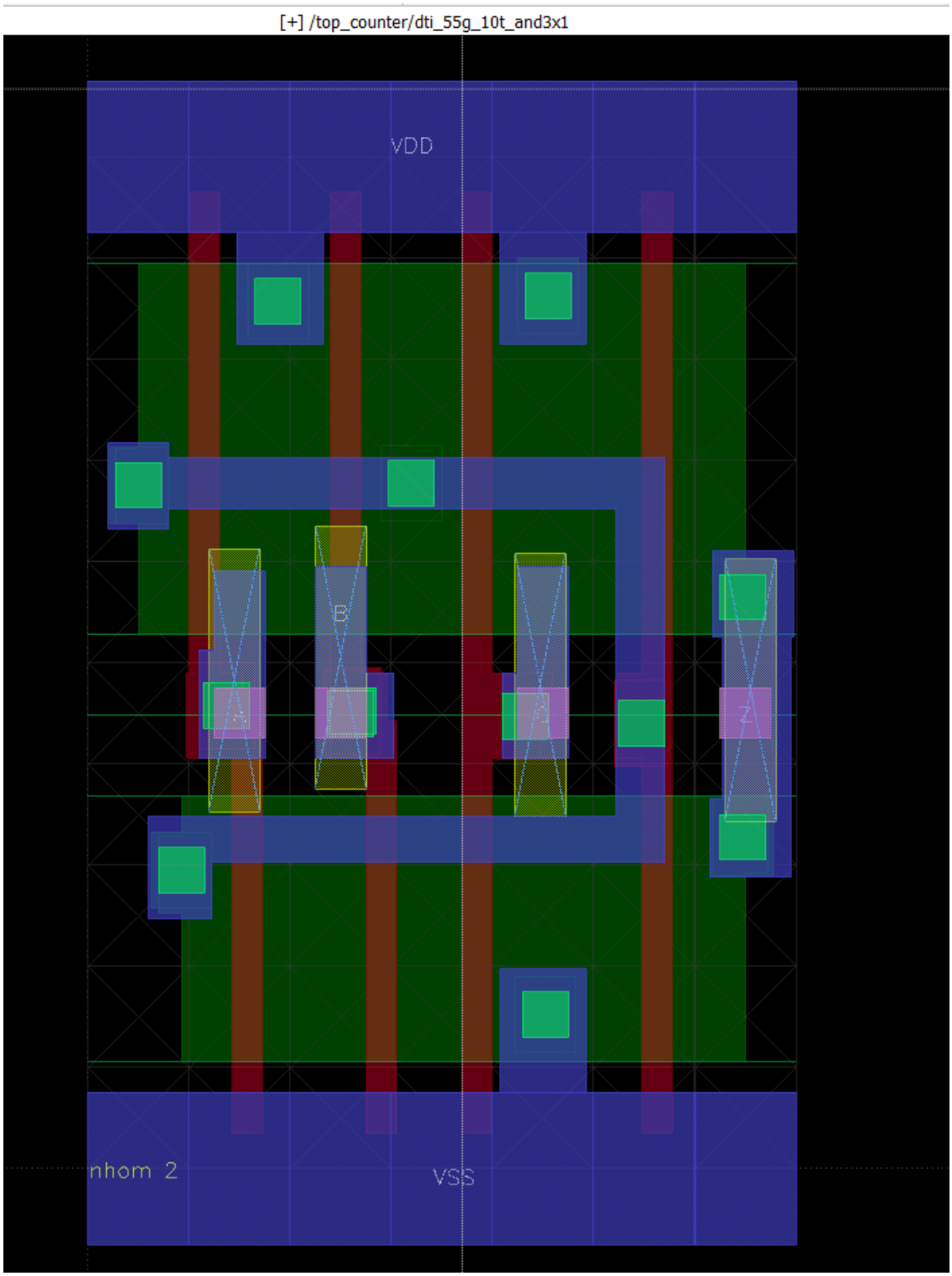
Hình 31. Hình kết quả 3



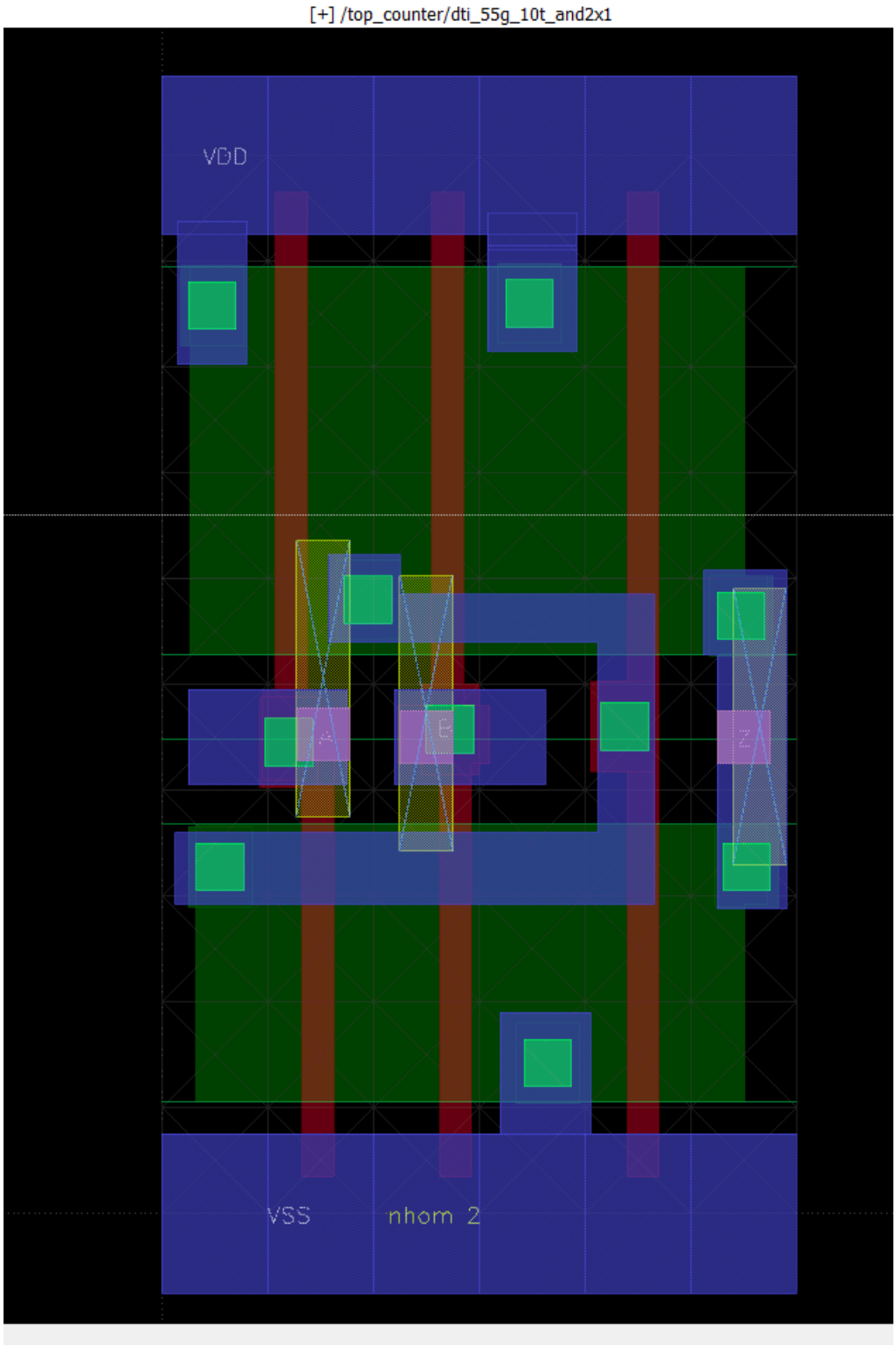
Hình 32. Hình kết quả 4



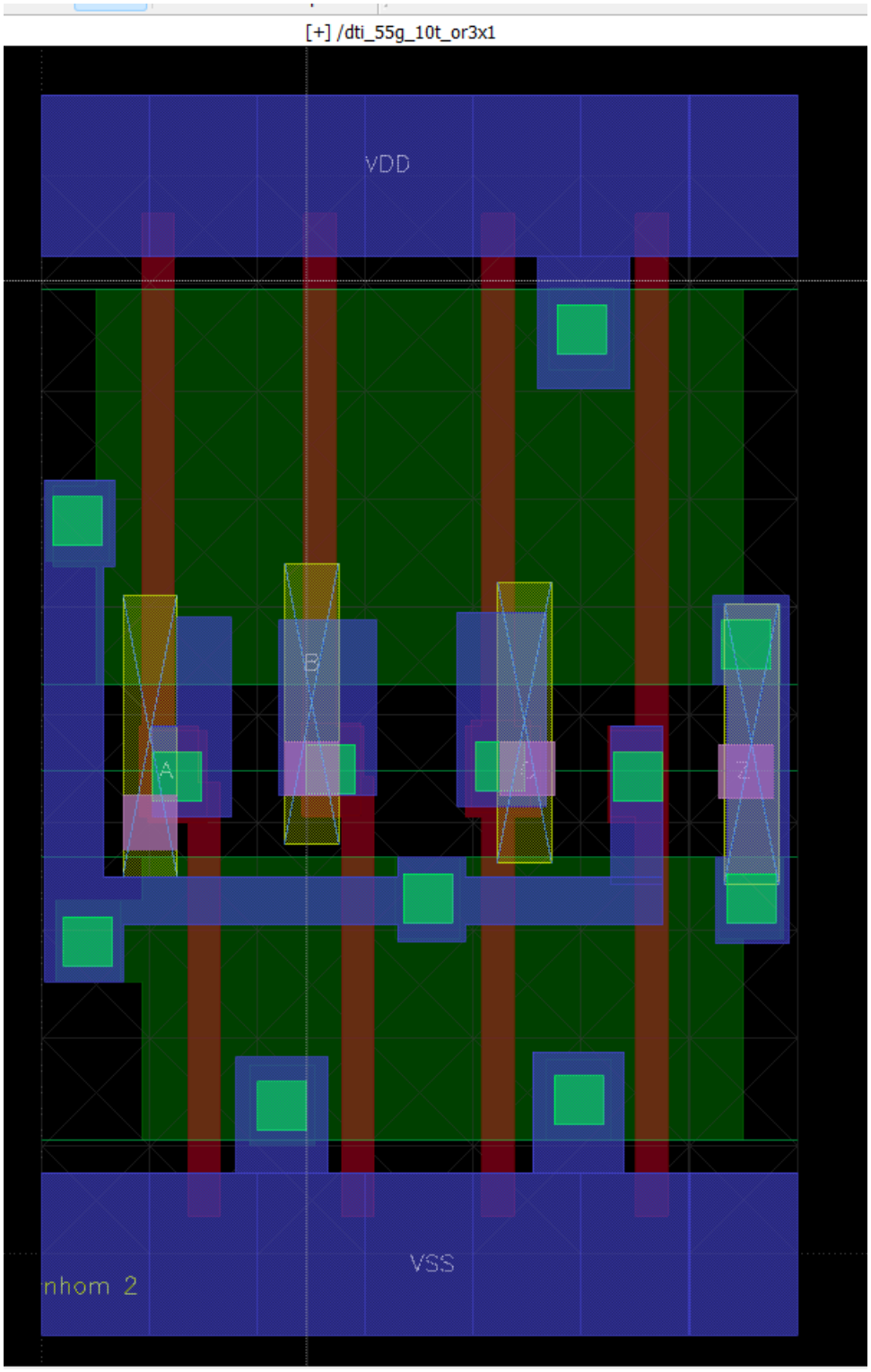
Hình 33. Hình kết quả 5



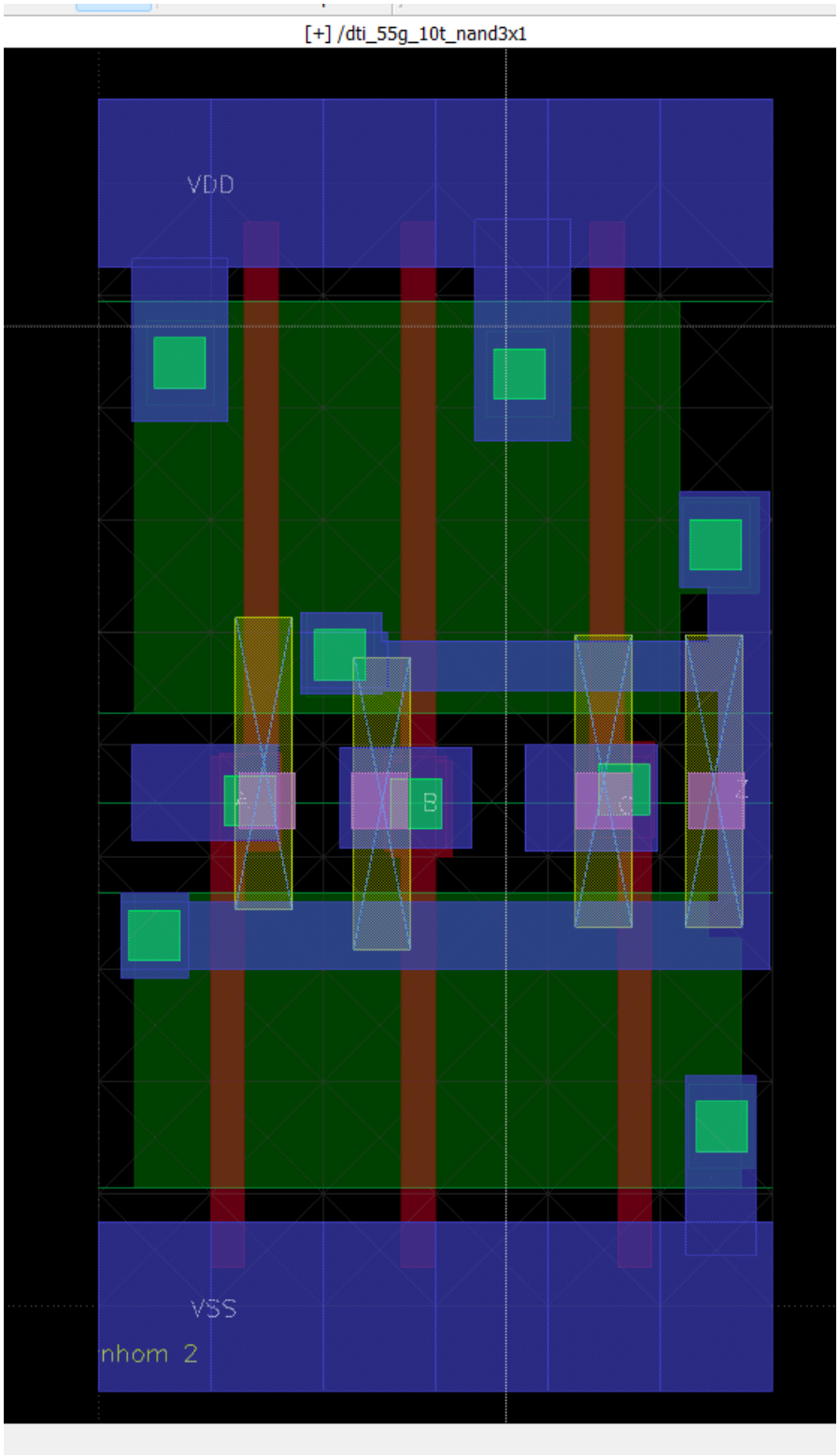
Hình 34. Hình kết quả 6



Hình 35. Hình kết quả 7



Hình 36. Hình kết quả 8



Hình 37. Hình kết quả 9

KẾT LUẬN

Trong báo cáo này, chúng em đã có cơ hội khám phá thế giới phức tạp và hết sức thú vị của vi mạch - nền tảng cơ bản của các thiết bị điện tử hiện đại.. Qua những bước phát triển quan trọng, chúng ta thấy rằng vi mạch đã không ngừng tiến xa và góp phần thay đổi cách chúng ta tương tác với công nghệ hàng ngày.

Cuối cùng, chúng em xin bày tỏ lòng cảm ơn chân thành đến thầy Nguyễn Vũ Thắng và các anh bên công ty Dolphin đã hướng dẫn và chia sẻ kiến thức và kinh nghiệm quý báu về đề tài này.. Chúng em hy vọng rằng báo cáo này đã đem lại những thông tin bổ ích và hữu ích cho bạn đọc, và chúng em sẵn sàng tiếp tục khám phá và đóng góp cho lĩnh vực thú vị này trong tương lai.