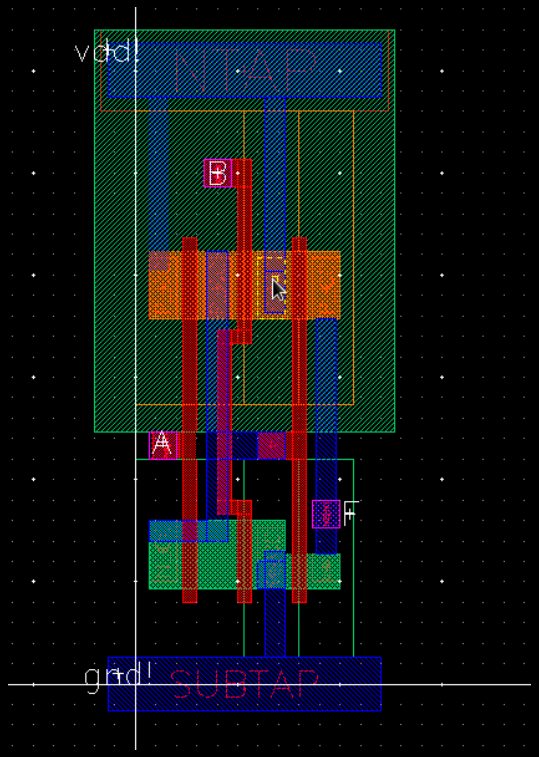
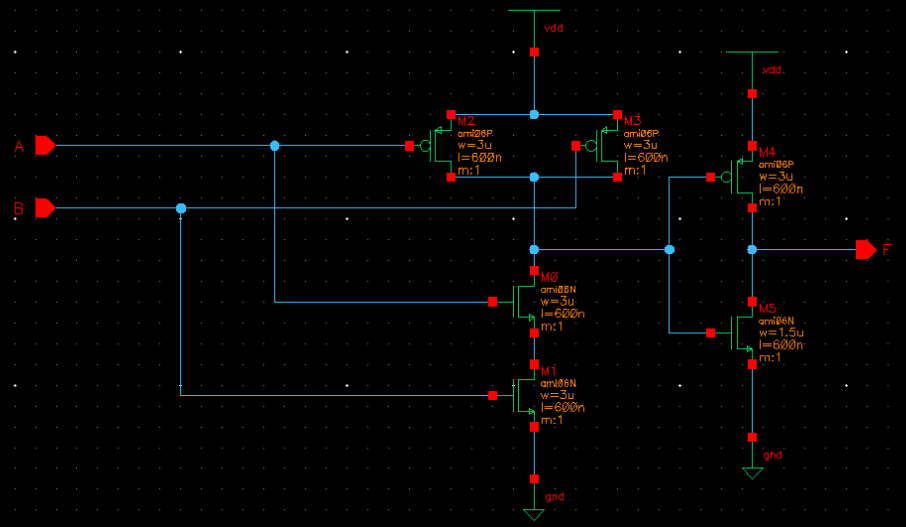
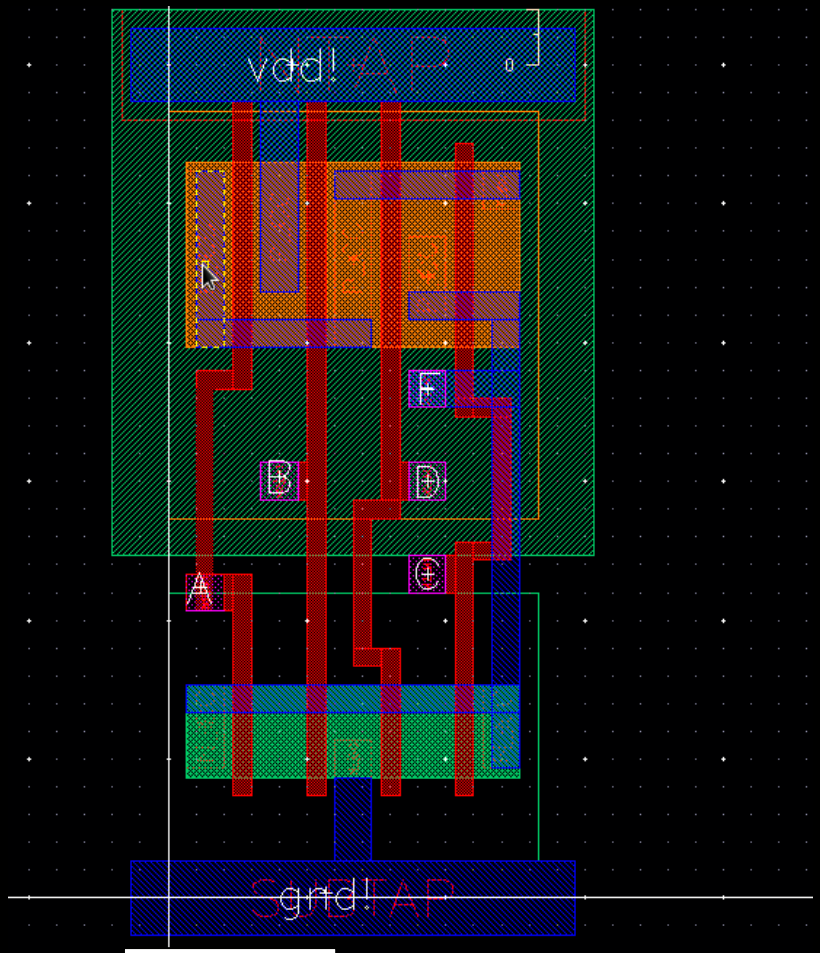


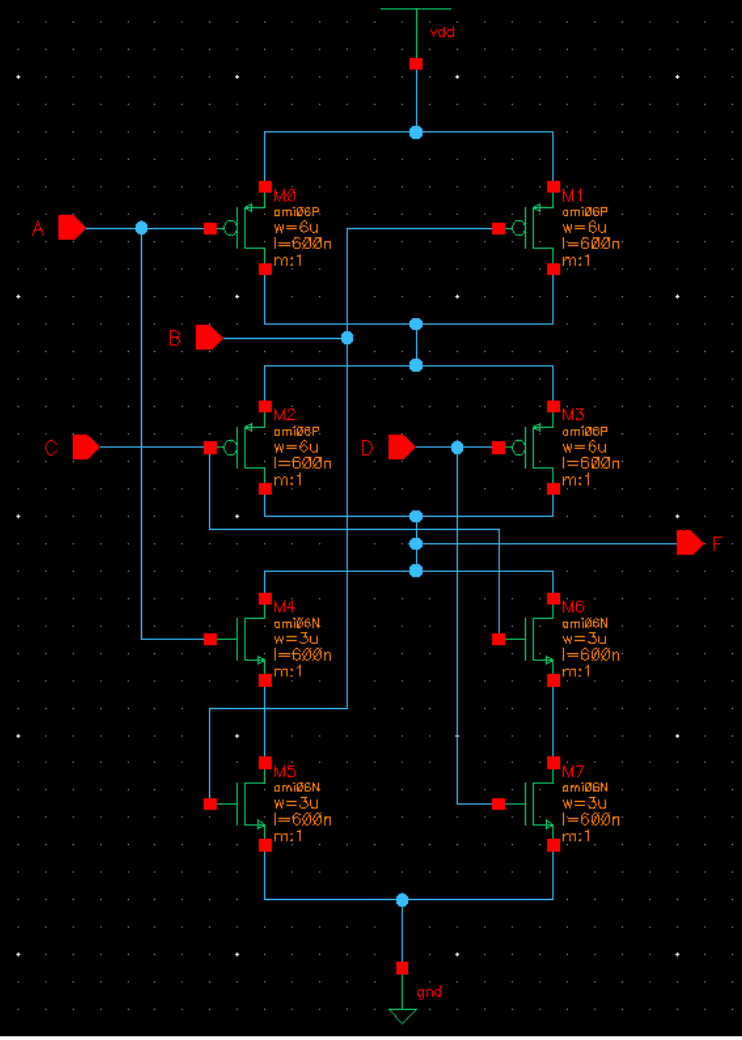
Area = 259.2

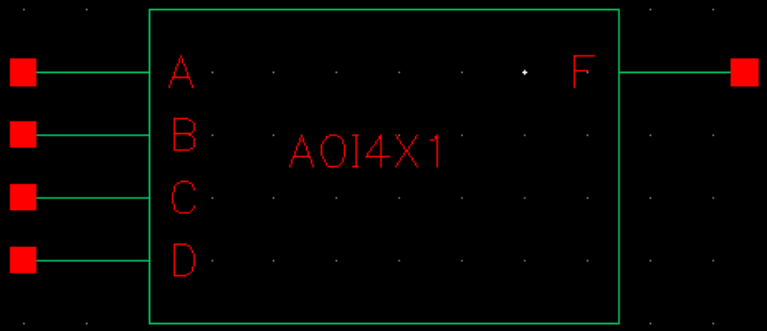



```
module AND2X1 ( F, A, B );

 output F;
 input A;
 input B;

 assign F = A & B;

 specify
    (A => F) = (1.0, 1.0);
    (B => F) = (1.0, 1.0);
 endspecify
```

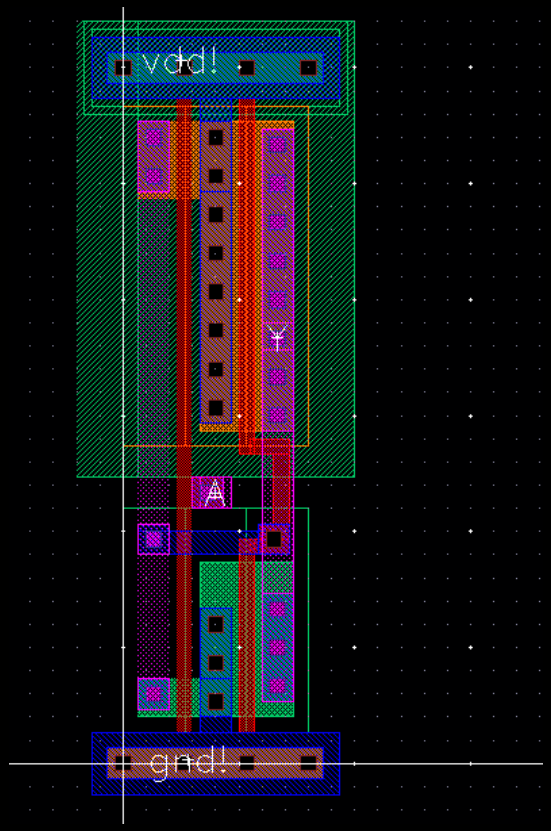


Area = 324

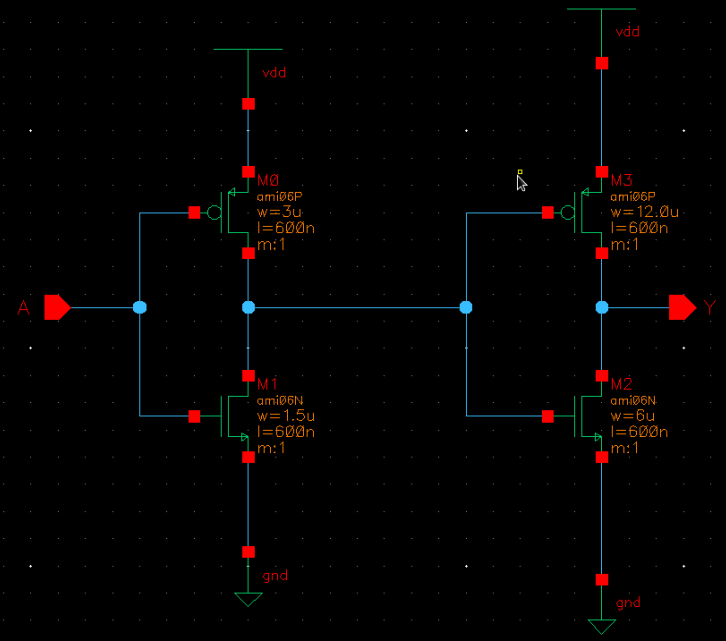




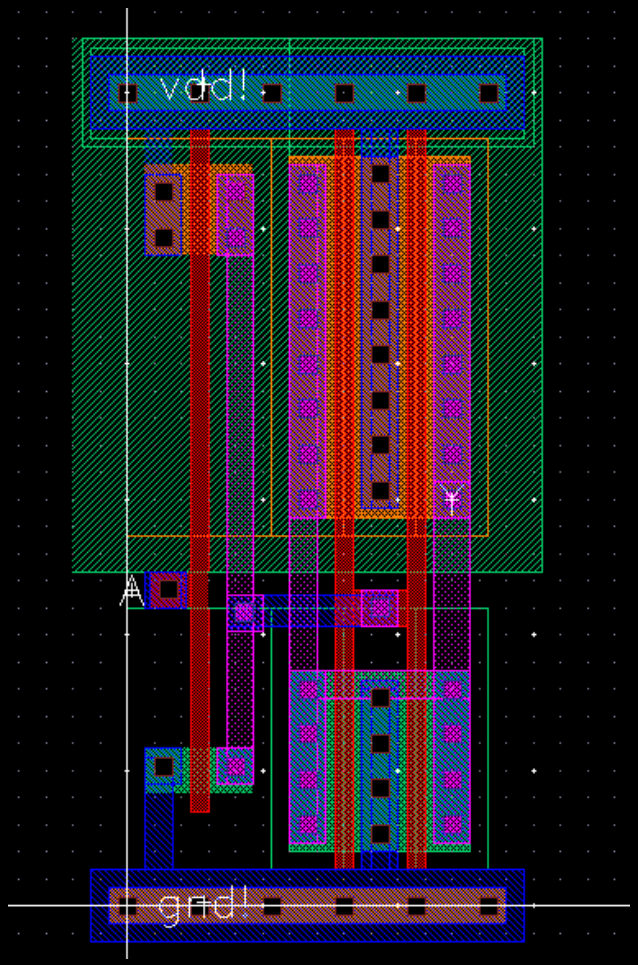

```
module AOI4X1 (F, A, B, C, D);
  output F;
  input A;
  input B;
  input C;
  input D;
  assign F = (!((A & B) | (C & D)));
  specify
    (A => F) = (1.0, 1.0);
    (B => F) = (1.0, 1.0);
    (C => F) = (1.0, 1.0);
    (D => F) = (1.0, 1.0);
  endspecify
  endmodule
```

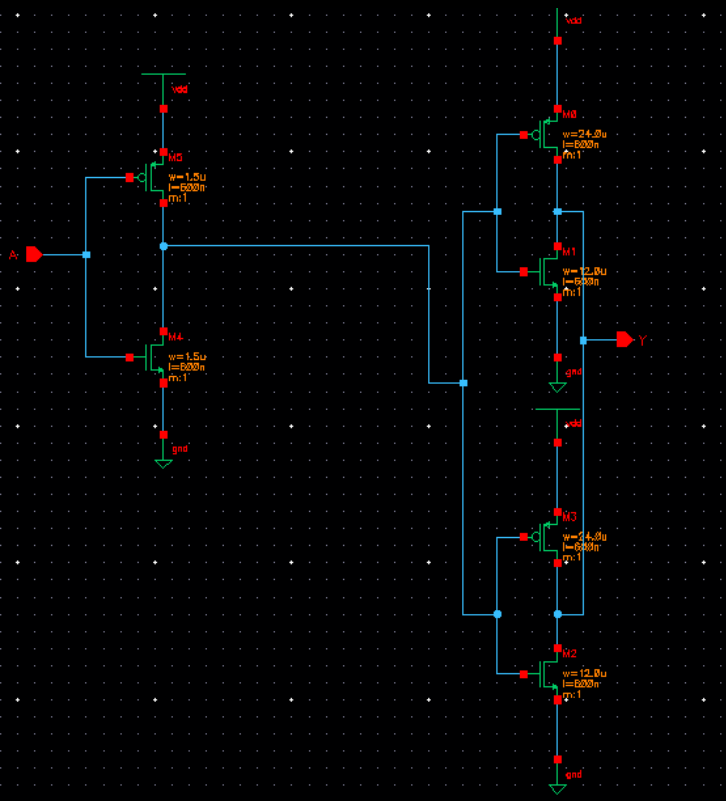

vdd!
A
Y
gnd!

Area = 194.4


vdd
M0
ami06P
w=3u
l=600n
m:1
M3
ami06P
w=12.0u
l=600n
m:1
A
Y
M1
ami06N
w=1.5u
l=600n
m:1
M2
ami06N
w=6u
l=600n
m:1
gnd
gnd

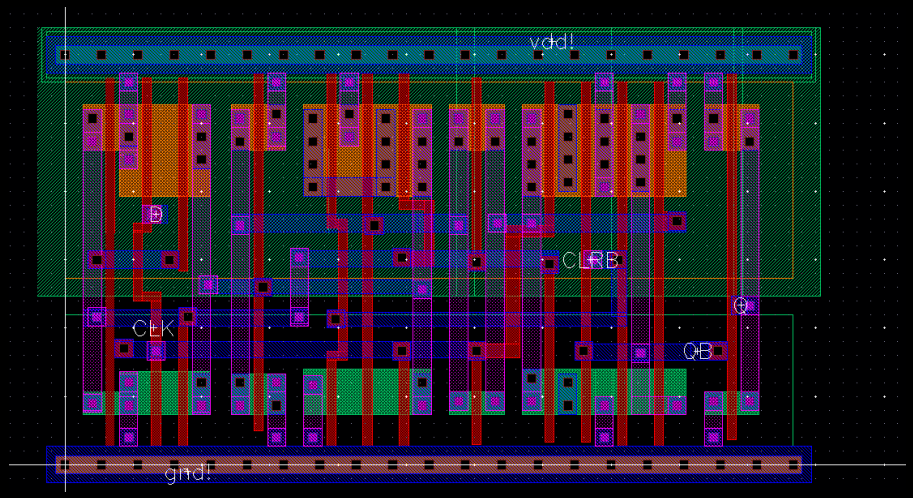

A
BUFX4
Y

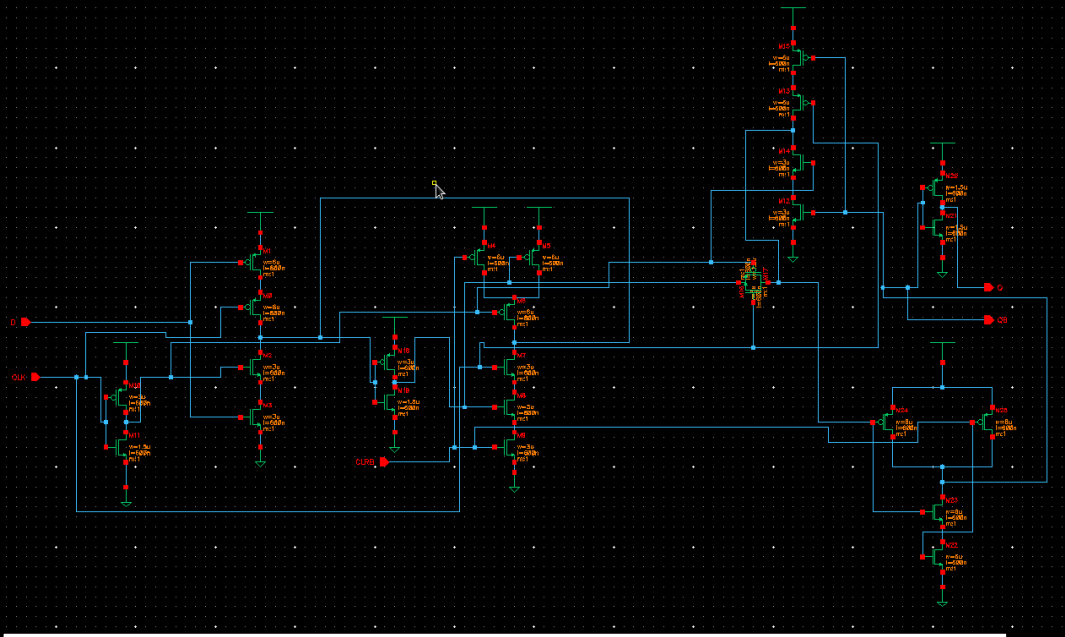

vdd!
A
Y
gnd!


vdd
M5
w=1.5u
l=600n
m:1
A
M4
w=1.5u
l=600n
m:1
gnd
M0
w=24.0u
l=600n
m:1
M1
w=12.0u
l=600n
m:1
Y
gnd
vdd
M3
w=24.0u
l=600n
m:1
M2
w=12.0u
l=600n
m:1
gnd

Area = 324

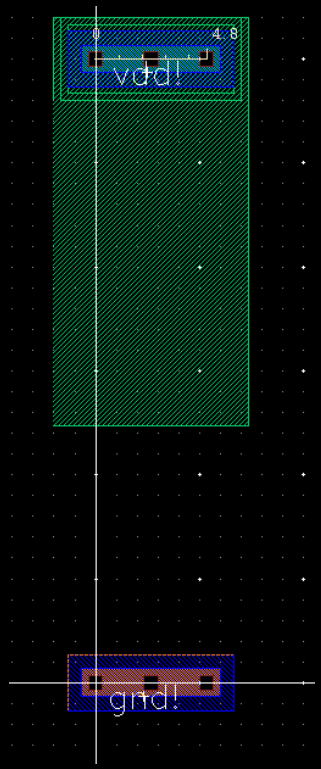

A
BUFX8
Y

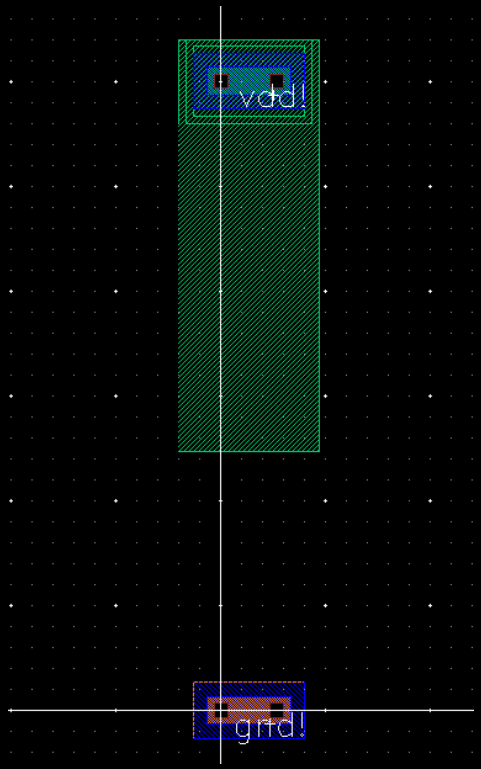


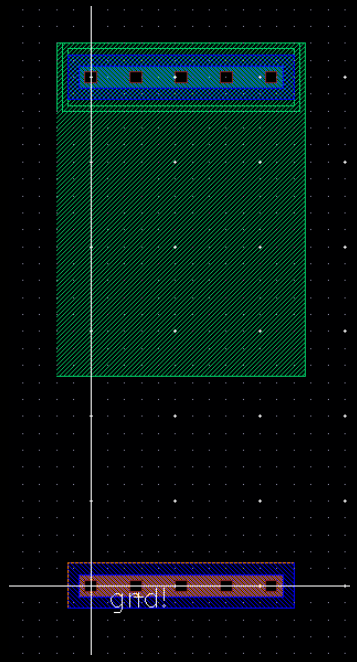


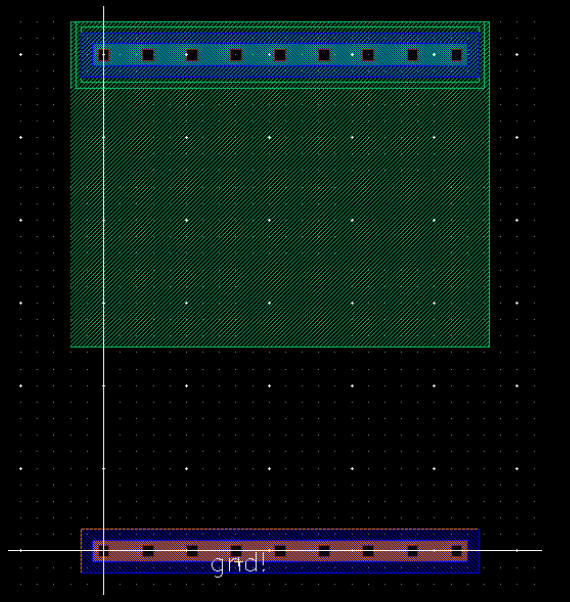
Area = 1296

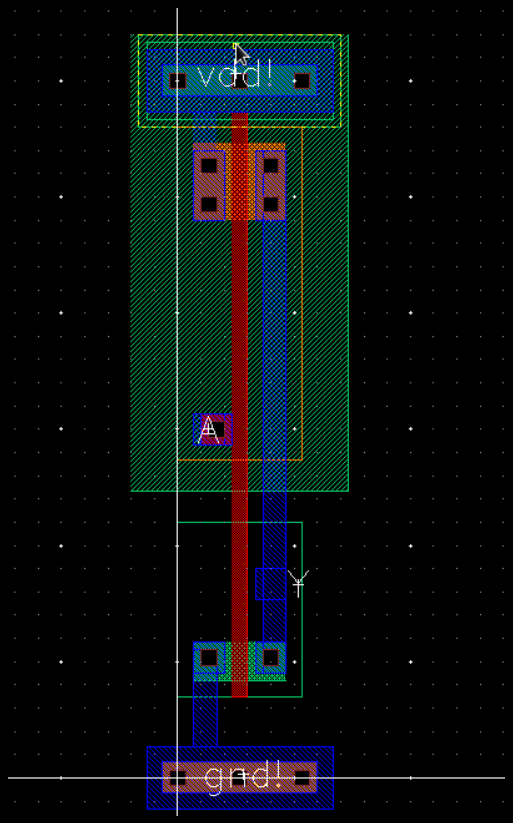

```
// rising edge triggered master slave DFF

module DFF( Q, QB, D, CLK, CLRB );
  output Q;
   output QB;
  input CLK;
  input D;
  input CLRB;
   reg     Q;
   always@(posedge CLK or negedge CLRB)
     if(!CLRB)
       Q <= 1'b0;
     else
       begin
       Q <= D;
      end
   specify
      (CLK => Q) = (1.0, 1.0);
      (CLRB => Q) = (1.0, 1.0);
      (D => Q) = (1.0, 1.0);
   endspecify
endmodule
```

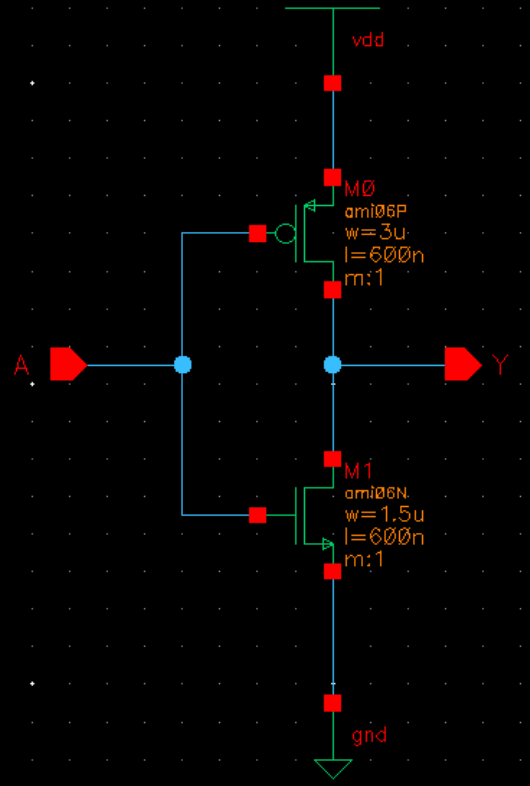

0
4.8
vdd!
gnd!


vdd!
gnd!

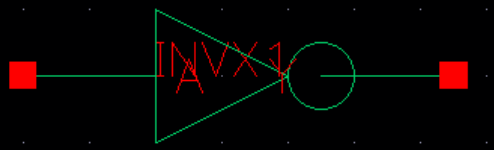

gnd!


gnd!

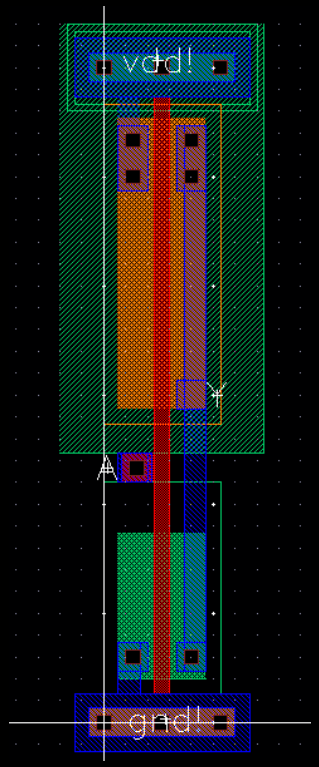


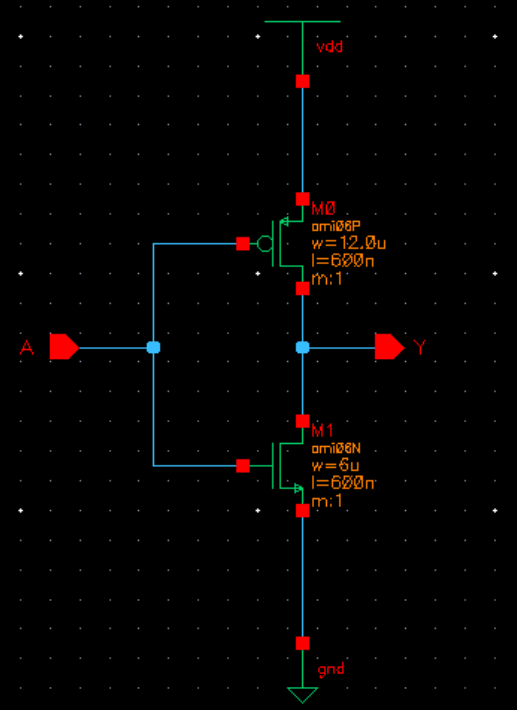


Area = 129.6



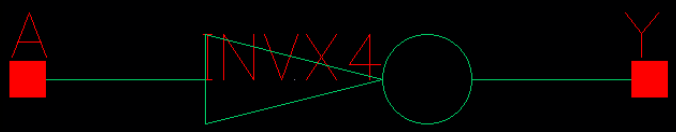

```
module INVX1 ( Y, A );

  input A;
  output Y;
   assign Y = ~A;

   specify
      (A => Y) = (1.0, 1.0);
   endspecify

endmodule
```

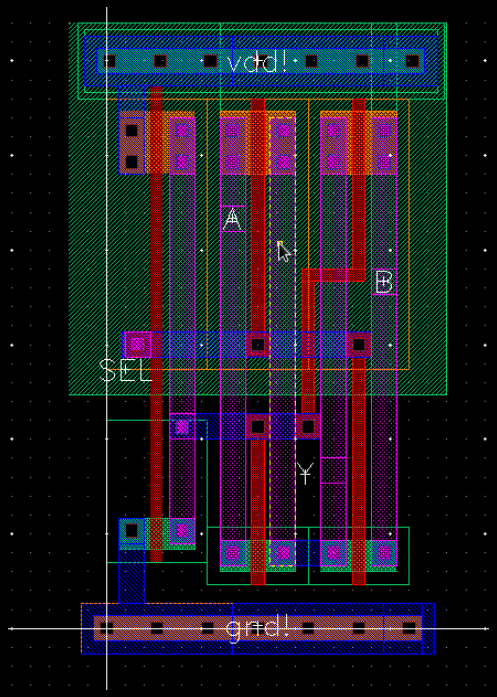


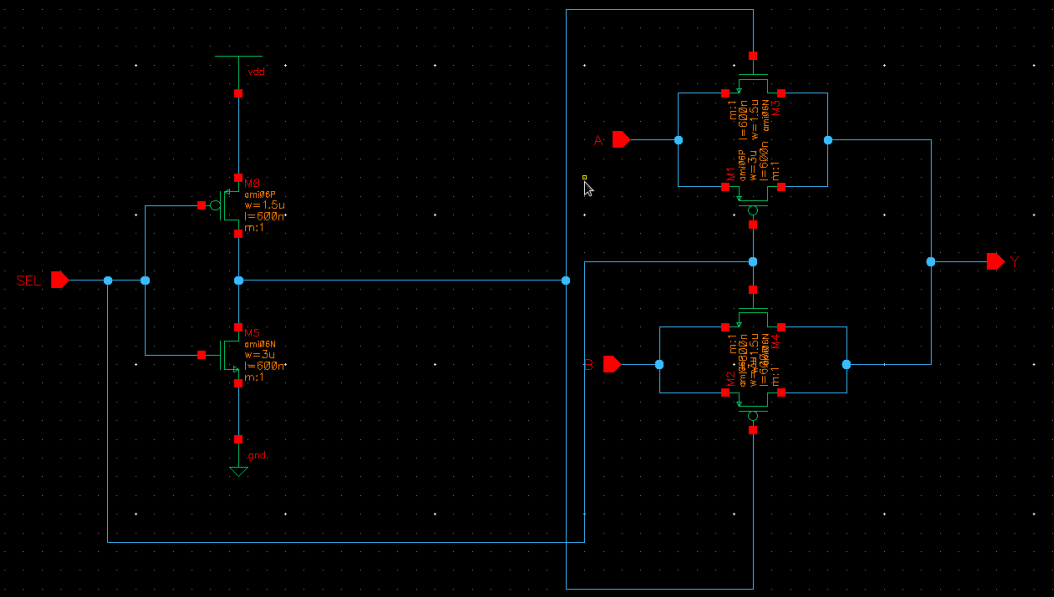


Area= 129.6

```
module INVX4 ( Y, A );

 input A;
 output Y;
  assign Y = ~A;

  specify
     (A => Y) = (1.0, 1.0);
  endspecify

endmodule
```

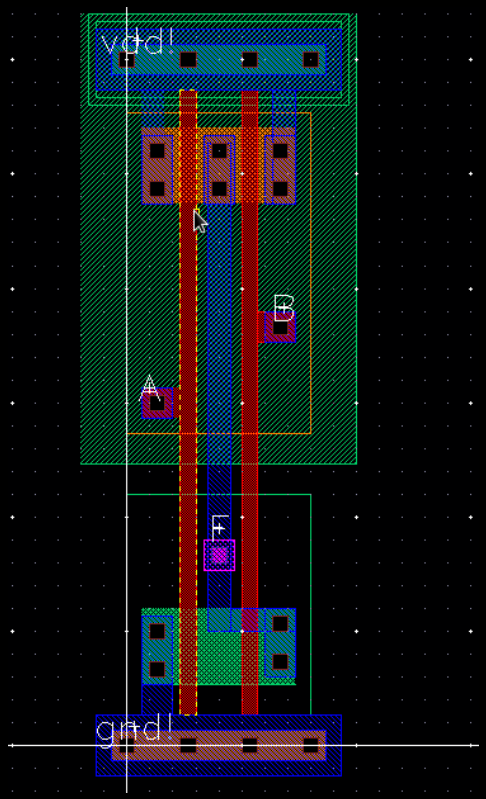


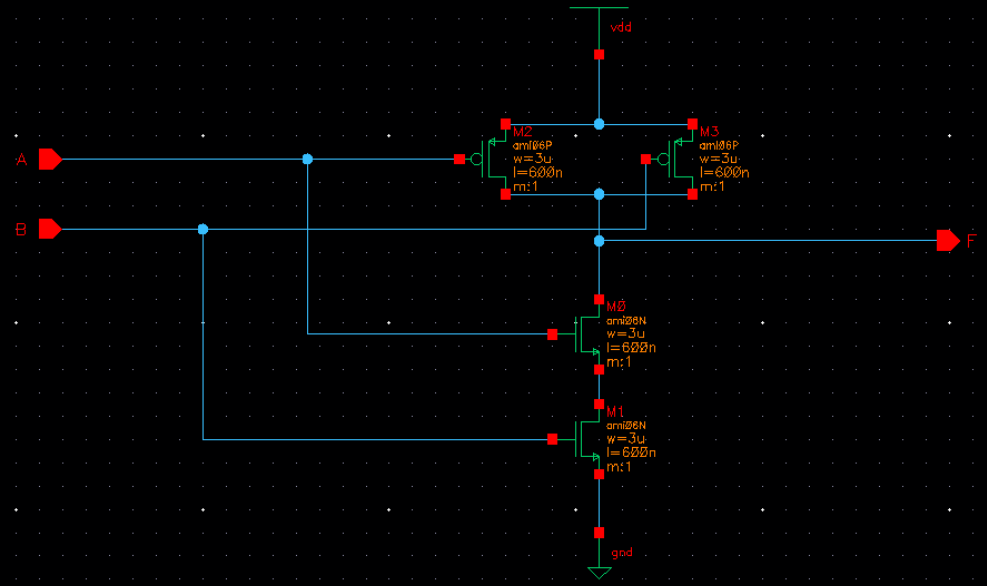




Area = 388.8

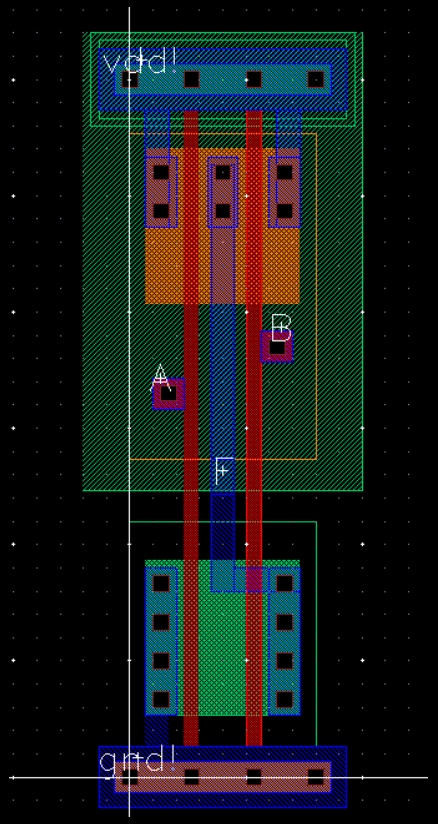

```
module MUX2X1( Y, A, B, SEL );

  input A;
  output Y;
  input SEL;
  input B;
   assign Y = (~SEL&A)|(SEL&B);
    specify
      (A => Y) = (1.0,1.0);
      (B => Y) = (1.0,1.0);
      (SEL => Y) = (1.0,1.0);
   endspecify
endmodule
```

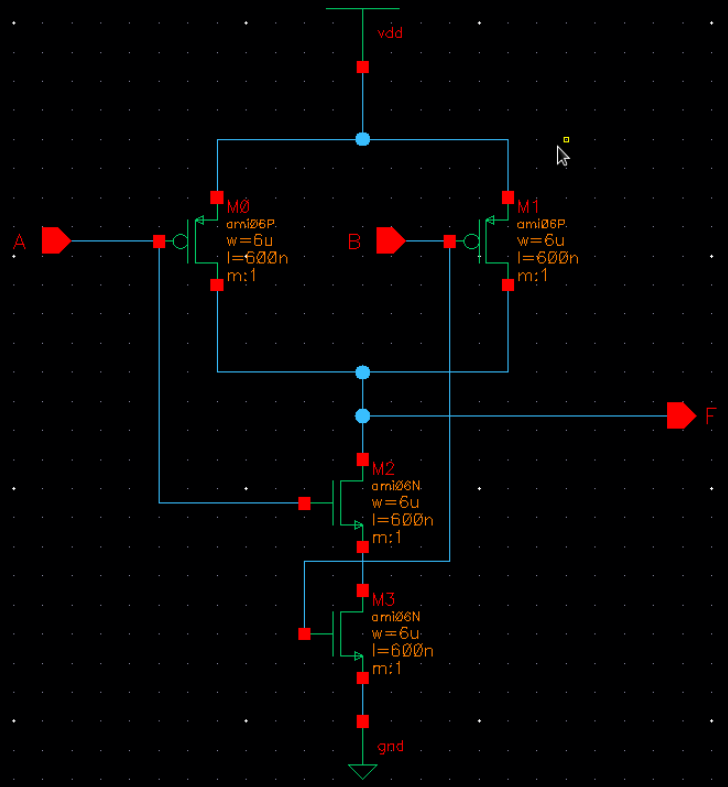




Area = 194.4

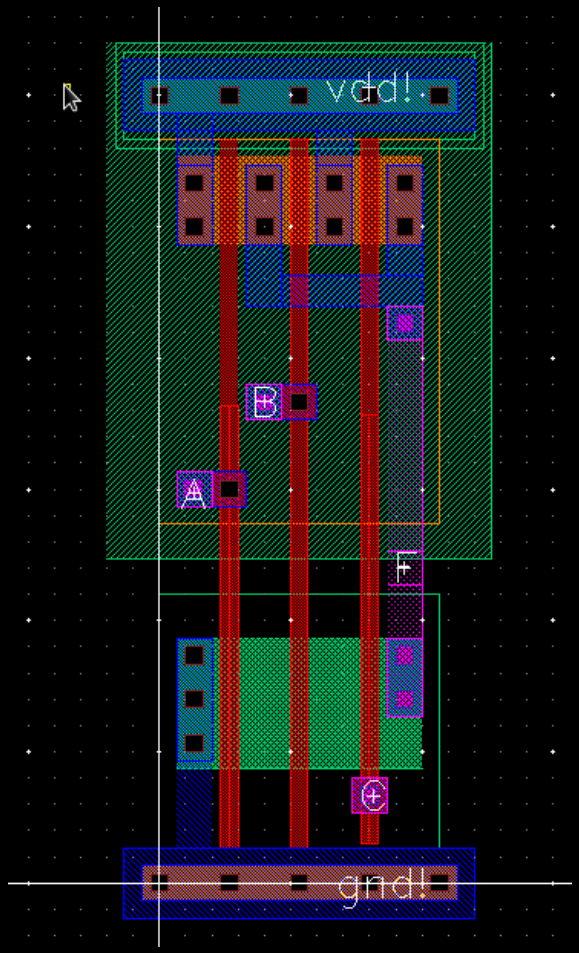
```
module NAND2X1 ( F, A, B );

  input A;
  output F;
  input B;
   assign F = ~(A & B);

   specify
      (A =>F) = (1.0, 1.0);
      (B =>F) = (1.0, 1.0);
   endspecify
endmodule
```

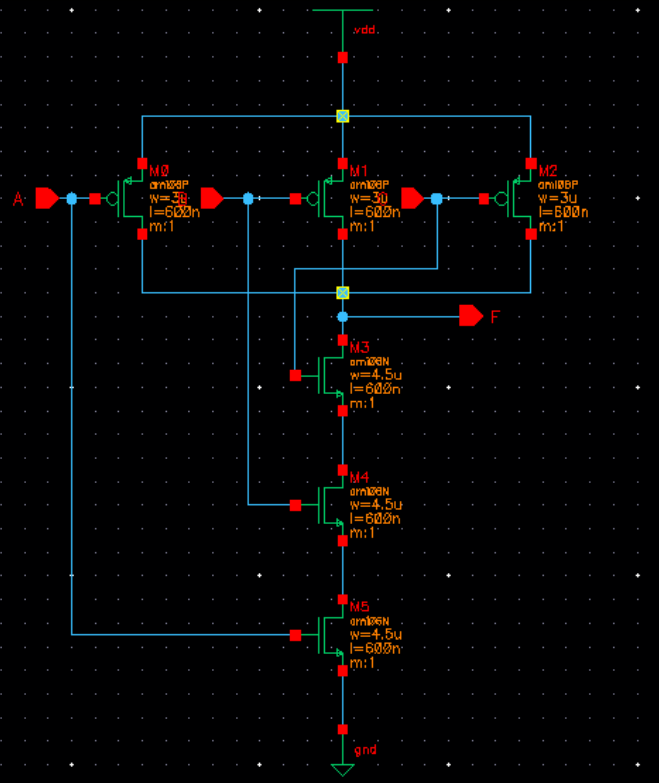

vdd!
B
A
F
gnd!


vdd
M0
ami06P
w=6u
l=600n
m:1
M1
ami06P
w=6u
l=600n
m:1
A
B
F
M2
ami06N
w=6u
l=600n
m:1
M3
ami06N
w=6u
l=600n
m:1
gnd

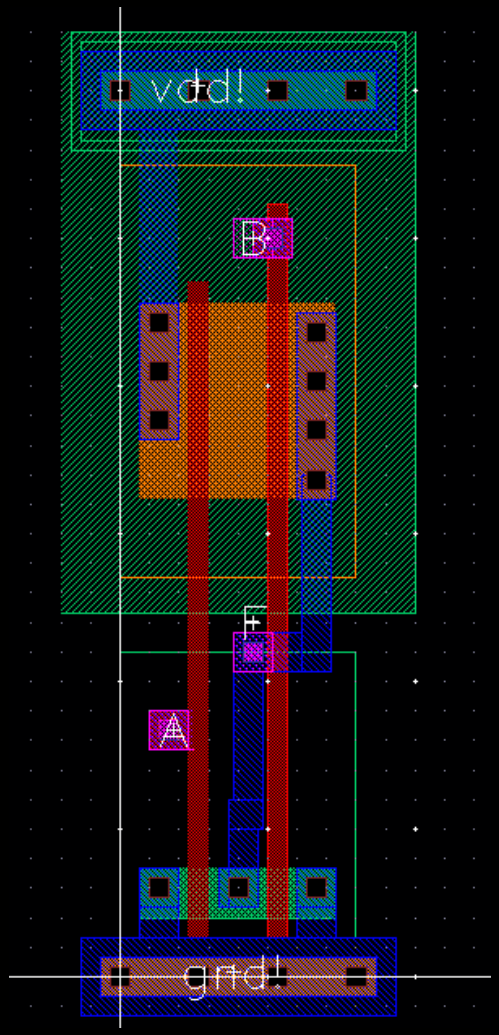
Area = 194.4

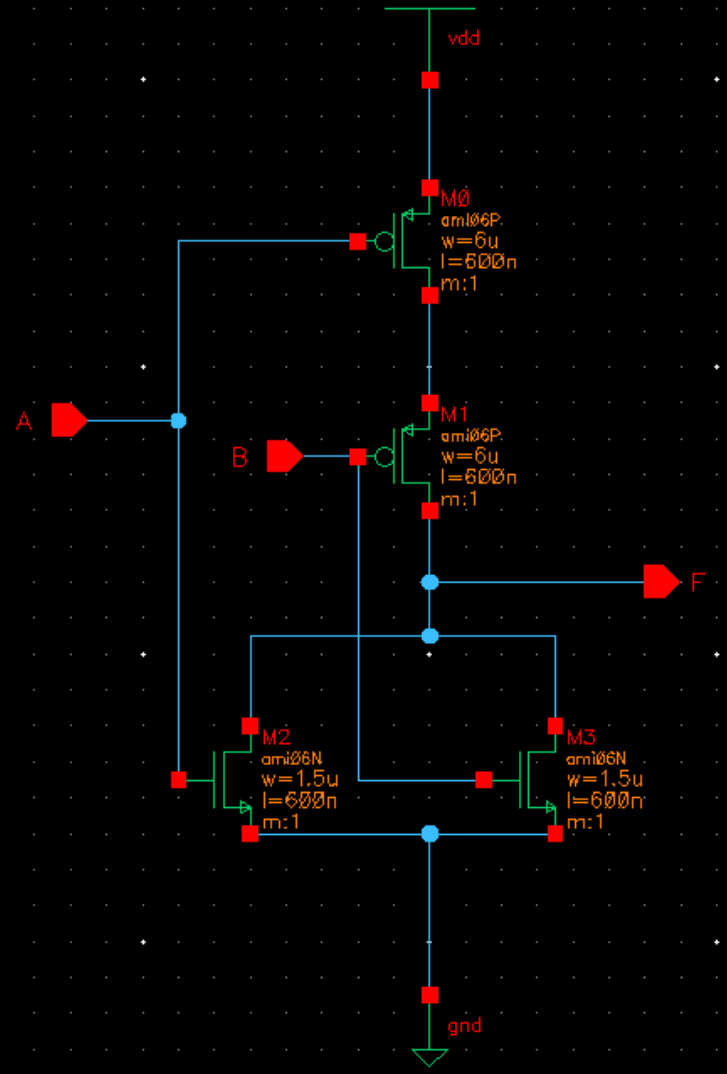

A
NAND2X2
B
F



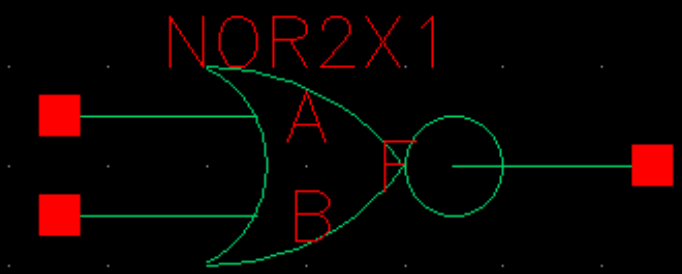


Area = 259.2

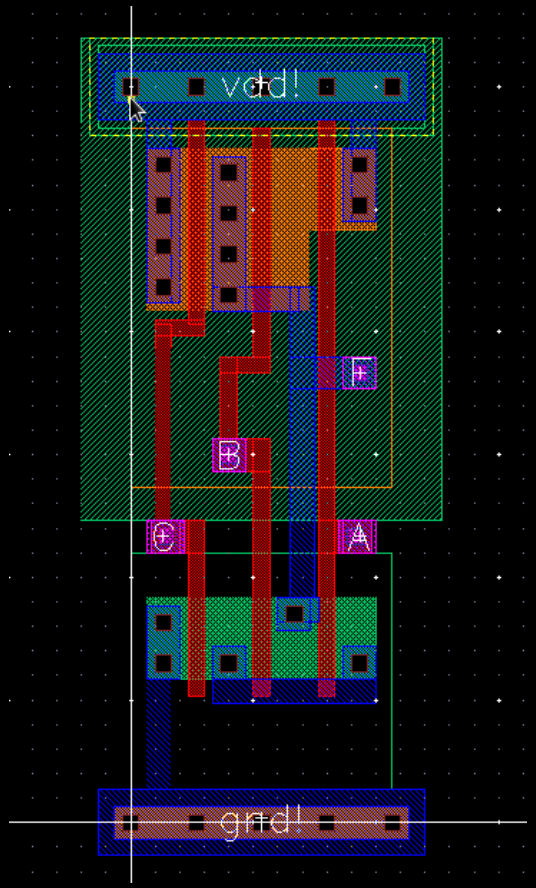


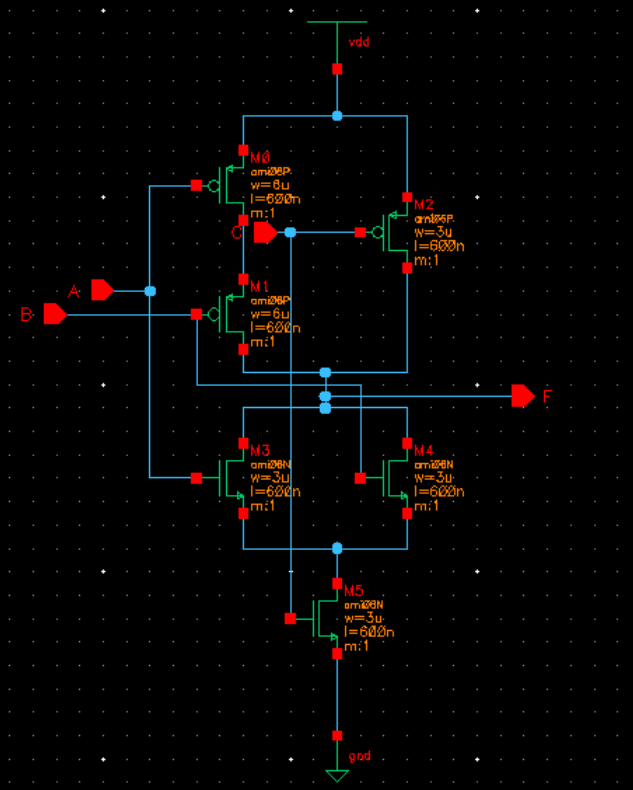


Area = 194.4



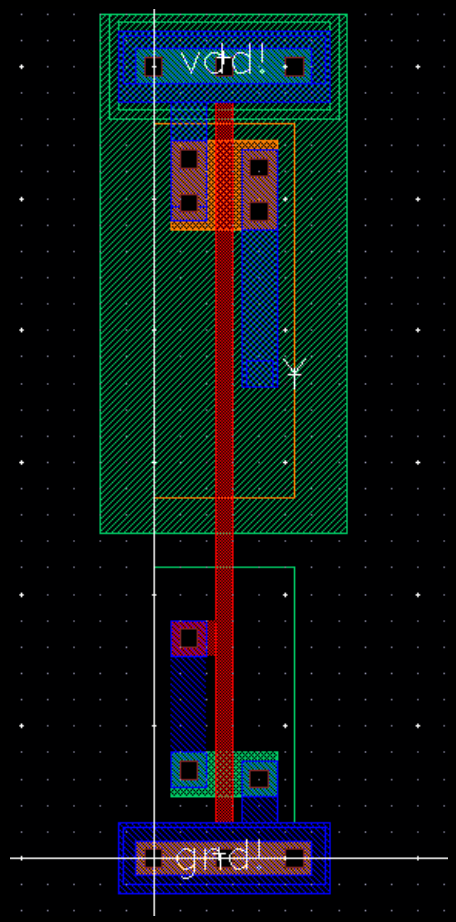

```
module NOR2X1 ( F, A, B );
  input A;
  output F;
  input B;
   assign F = ~(A | B);
   specify
      (A => F) = (1.0, 1.0);
      (B => F) = (1.0, 1.0);
   endspecify
endmodule
```

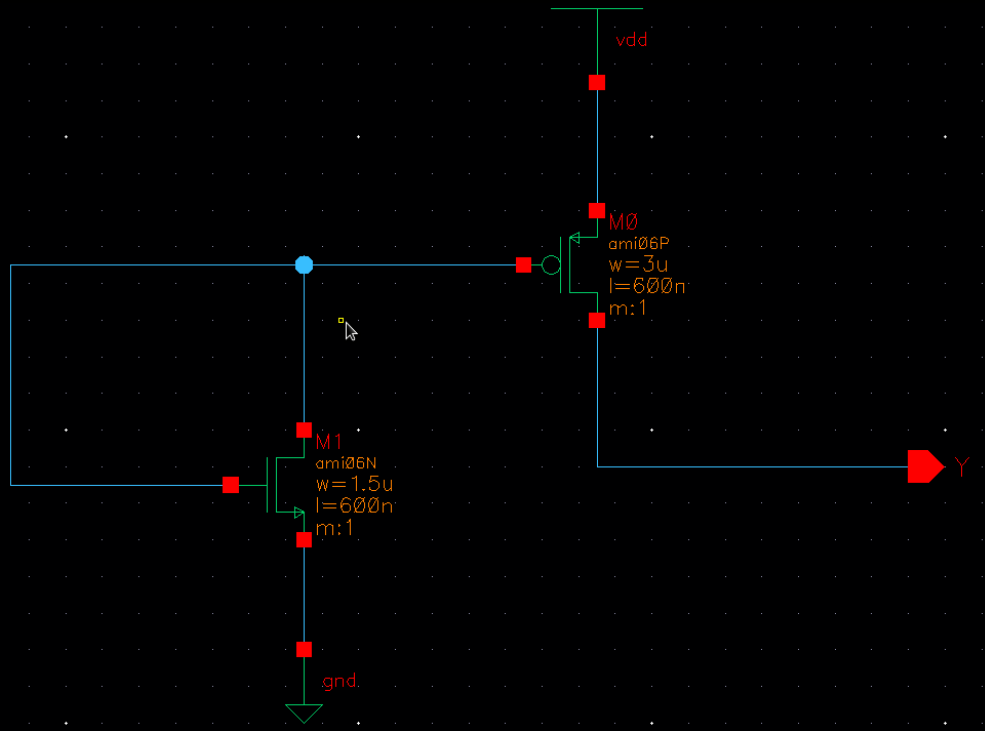


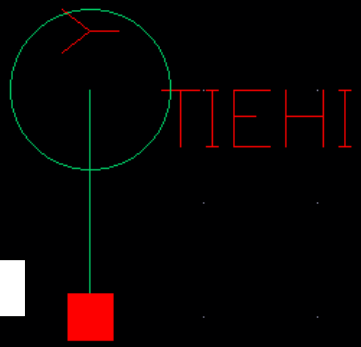


Area = 259.2

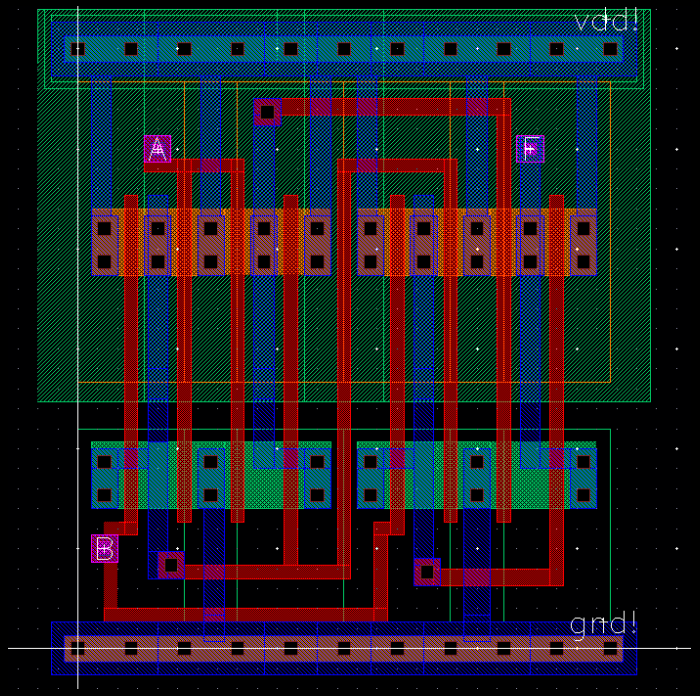

```
module OAI3X1 ( F, A, B, C );
  input A;
  output F;
  input B;
  input C;
  assign F = !(C & (A | B));
  specify
    (A => F) = (1.0, 1.0);
    (B => F) = (1.0, 1.0);
    (C => F) = (1.0, 1.0);
  endspecify
endmodule
```

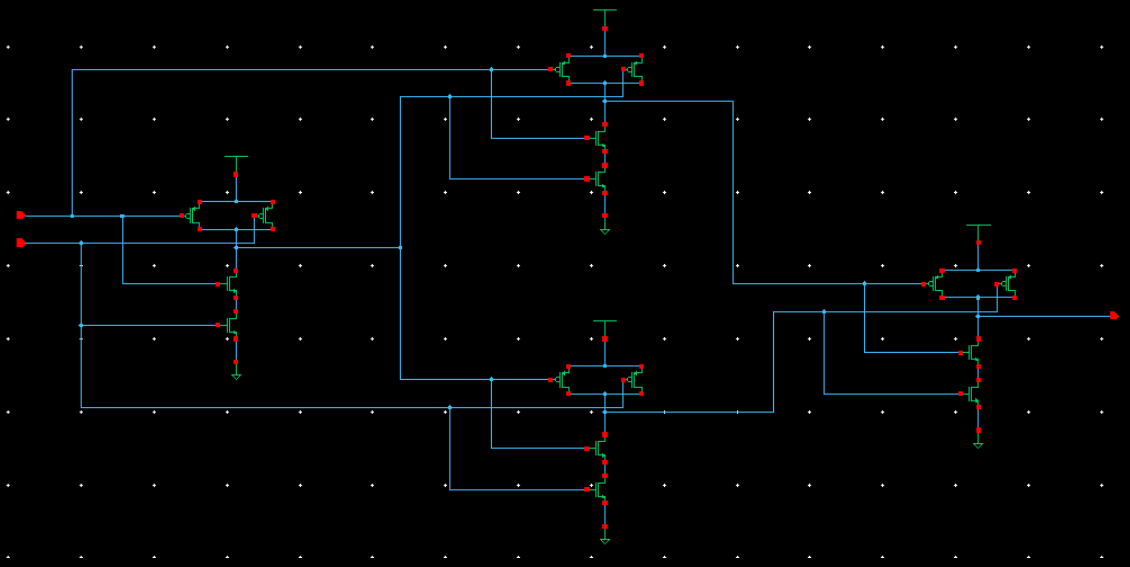

vdd!
gnd!

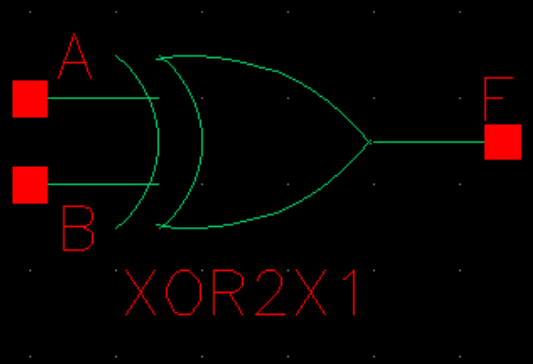

vdd
MØ
amiØ6P
w=3u
l=6ØØn
m:1
M1
amiØ6N
w=1.5u
l=6ØØn
m:1
Y
gnd


TIEHI

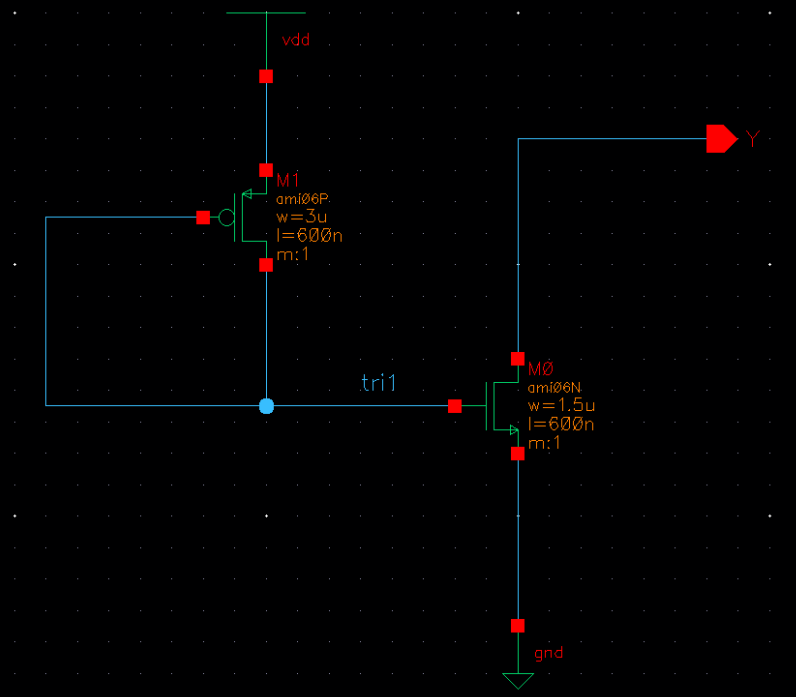
Area = 648

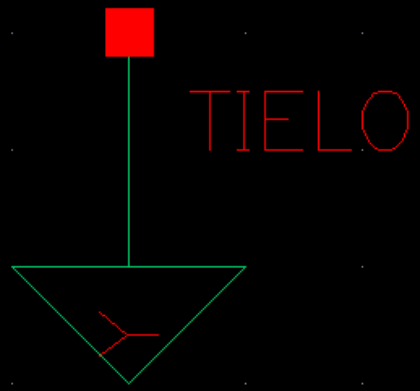







```
module XOR ( F, A, B );

  input A;
  output F;
  input B;

   assign F = A ^ B;

   specify
      (A => F) = (1.0, 1.0);
      (B => F) = (1.0, 1.0);
   endspecify

endmodule
```


vdd
M1
ami06P
w=3u
l=600n
m:1
Y
tri1
M0
ami06N
w=1.5u
l=600n
m:1
gnd


TIELO