Elec 422: VLSI Design I

Notes on IRSIM Test Vectors

Modified Up-Down Counter with Hold

The counter example presented in the PLANOTE can be modified to add a forward and back feature along with a hold feature. Note that the RESTART signal does not need to be watched in the *case* statements. The "RESET on" statement has global scope. The meg file becomes:

```
--      MEG program
--      2-bit counter,  0,3,1,2
--      with even/odd output
--      state transition on every clock
--      Forward, Back counts, and Hold.

INPUTS:  RESTART HOLD FORWARD BACK;
OUTPUTS: ODD MSB LSB;

RESET ON RESTART TO state0;

state0  :       case (HOLD FORWARD BACK)
                       0     1      0    => state3 (ODD=1 MSB=1 LSB=1);
                       0     0      1    => state2 (ODD=0 MSB=1 LSB=0);
                       1     0      0    => state0 (ODD=0 MSB=0 LSB=0);
                endcase => ANY;

state1  :       case (HOLD FORWARD BACK)
                       0     1      0    => state2 (ODD=0 MSB=1 LSB=0);
                       0     0      1    => state3 (ODD=1 MSB=1 LSB=1);
                       1     0      0    => state1 (ODD=1 MSB=0 LSB=1);
                endcase => ANY;


state2  :       case (HOLD FORWARD BACK)
                       0     1      0    => state0 (ODD=0 MSB=0 LSB=0);
                       0     0      1    => state1 (ODD=1 MSB=0 LSB=1);
                       1     0      0    => state2 (ODD=0 MSB=1 LSB=0);
                endcase => ANY;

state3  :       case (HOLD FORWARD BACK)
                       0     1      0    => state1 (ODD=1 MSB=0 LSB=1);
                       0     0      1    => state0 (ODD=0 MSB=0 LSB=0);
                       1     0      0    => state3 (ODD=1 MSB=1 LSB=1);
                endcase => ANY;
```

This produces the following *meg.summary* file and *magic* layout:

```
SUMMARY INFORMATION GENERATED BY MEG FROM FILE counter.meg
meg:  Number of inputs  =4
meg:  Number of outputs =3
meg:  Number of states  =4
meg:  State field size  =2 * 2
meg:  Table width       =11


INPUTS:
   i00:  RESTART
   i01:  HOLD
   i02:  FORWARD
   i03:  BACK
   s00:  StBit0* (msb)
   s01:  StBit1* (lsb)


OUTPUTS:
   n01:  StBit1* (lsb)
   n00:  StBit0* (msb)
   o00:  ODD
   o01:  MSB
   o02:  LSB


State Table
     i  i  i  i  s  s  n  n  o  o  o
     0  1  2  3  0  1  1  0  0  1  2
   ---------------------------------
     1  -  -  -  -  -  0  0  0  0  0   ANY

     0  0  0  0  0  0  -  -  0  0  0   state0
     0  1  0  0  0  0  0  0  0  0  0   state0
     0  0  1  0  0  0  1  1  1  1  1   state0
     0  1  1  0  0  0  -  -  0  0  0   state0
     0  0  0  1  0  0  0  1  0  1  0   state0
     0  1  0  1  0  0  -  -  0  0  0   state0
     0  0  1  1  0  0  -  -  0  0  0   state0
     0  1  1  1  0  0  -  -  0  0  0   state0

     0  0  0  0  0  1  -  -  0  0  0   state1
     0  1  0  0  0  1  1  0  1  0  1   state1
     0  0  1  0  0  1  0  1  0  1  0   state1
     0  1  1  0  0  1  -  -  0  0  0   state1
     0  0  0  1  0  1  1  1  1  1  1   state1
     0  1  0  1  0  1  -  -  0  0  0   state1
     0  0  1  1  0  1  -  -  0  0  0   state1
```

```
0  1  1  1  0  1  -  -  0  0  0   state1

0  0  0  0  1  0  -  -  0  0  0   state2
0  1  0  0  1  0  0  1  0  1  0   state2
0  0  1  0  1  0  0  0  0  0  0   state2
0  1  1  0  1  0  -  -  0  0  0   state2
0  0  0  1  1  0  1  0  1  0  1   state2
0  1  0  1  1  0  -  -  0  0  0   state2
0  0  1  1  1  0  -  -  0  0  0   state2
0  1  1  1  1  0  -  -  0  0  0   state2

0  0  0  0  1  1  -  -  0  0  0   state3
0  1  0  0  1  1  1  1  1  1  1   state3
0  0  1  0  1  1  1  0  1  0  1   state3
0  1  1  0  1  1  -  -  0  0  0   state3
0  0  0  1  1  1  0  0  0  0  0   state3
0  1  0  1  1  1  -  -  0  0  0   state3
0  0  1  1  1  1  -  -  0  0  0   state3
0  1  1  1  1  1  -  -  0  0  0   state3


------------------------------
meg:  Eliminated Vectors=359568
```

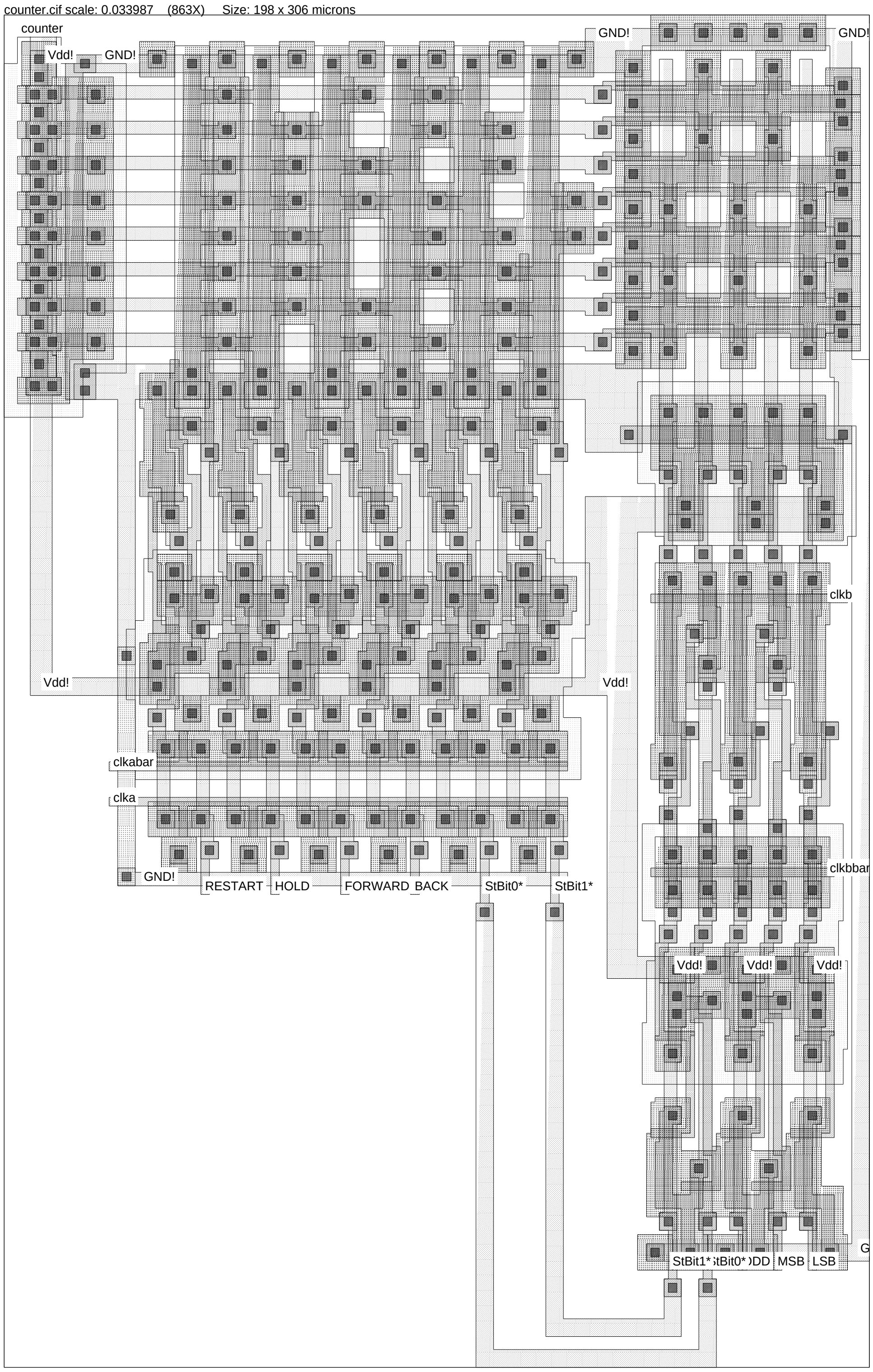

Figure 1: Magic layout after connection of Vdd and state bits.

Counter Simulation with IRSIM

In this simulation, several features are added in the command files. First, there is nesting of the command file *reset.cmd* within the main command file *counter.cmd*. This allows you to build specialized test sequence "subroutines" and include them as needed. Also, the "vector" command is used to watch the state bits and the "control" input. The "set" command is used alternatively with the "V" command to provide the control input. Observe the timing of the output signals with respect to the two-phase clock signals.

The main command file is:

```
| IRSIM test vector file for counter
|

clock CLKA    0 1 0 0
clock CLKABAR 1 0 1 1
clock CLKB    0 0 0 1
clock CLKBBAR 1 1 1 0

vector STATE StBit{0:1}*
vector control HOLD FORWARD BACK

ana CLKA CLKB StBit0* StBit1* STATE RESTART HOLD FORWARD control
ana BACK ODD MSB LSB

| read in sub-file that performs a specific function, like reset.
@ reset.cmd

| 16 clock cycles
V RESTART 0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
V control 010 010 100 100 010 010 010 010 010 100 100 001 001 001 001 001
R

| 2 clock cycles
| clear vectors before set command used
V
V RESTART 0 0
set control 010
R

| 2 clock cycles
| next section of test vectors that set the control as a vector.

V RESTART 1   0
V control 100 100
R
```

The sub-file $reset.cmd$ contains:

```
| IRSIM test vector sub-file to reset counter PLA
|

V RESTART 0   1   0
V control 000 000 000
R
```

The results of the $irsim$ simulation appear on the following page:

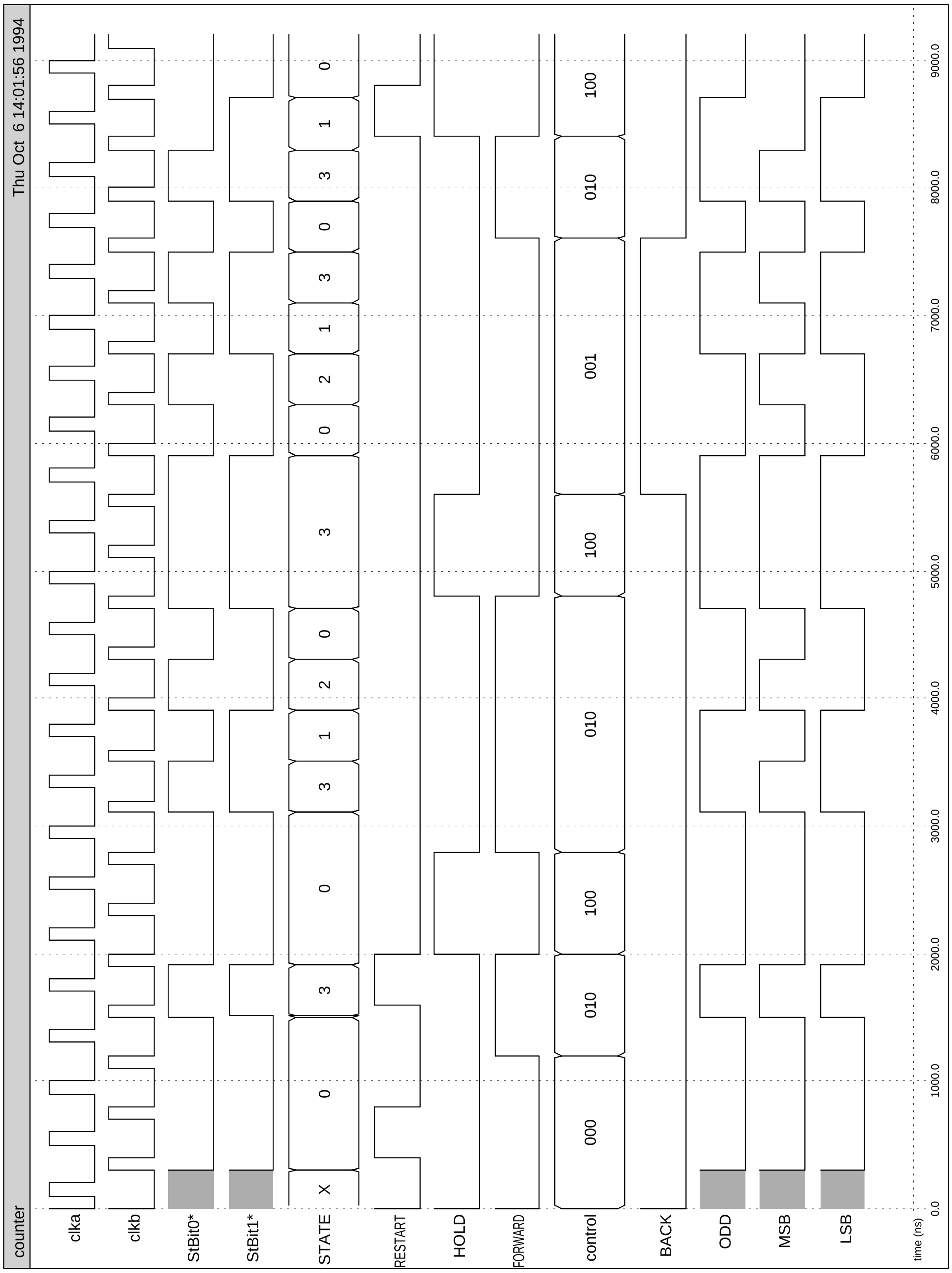

Figure 2: IRSIM simulation of Up-Down counter PLA.