Elec 422: VLSI Design I

Notes on PLA Tools

Fall 1997, October 2, 1997

PLATOOL Shell Script

Use the *platool* shell script to design the PLA's in your projects. It uses the most recent PLA template, /usr/site/cad/lib/mpla/p-SCS3cisnew.mag to build the output *magic* layout. The contents of /usr/site/cad/bin/platool are as follows:

```
#! /bin/csh -f
# usage for meg

meg -s -t -e $1.meg > $1.eqn
eqntott -l -f $1.eqn | espresso | mpla -I -O -s SCS3cisnew > $1.mag
```

Note that the usage is *platool file* where file is the filename without extensions. The shell script then looks for *file.meg* and produces a *magic* layout file *file.mag*. The default PLA template p-SCS3cisnew.mag can be viewed through *magic*. Sections of it are assembled by *mpla* to form the layout. This template provides clocked latches at both the inputs and outputs.

MPLA Limitations

Vdd Connection: The clocked outputs of the PLA require Vdd to be attached to the main PLA Vdd. (All Vdd segments are labeled.) See the attached plot. **It is vital to connect Vdd to the output registers!**

Substrate Error: Large PLA's may have design rule errors around the substrate contact in the upper left corner of the PLA. Use *magic* to enlarge the substrate contact to fix the design rule errors.

Feedback Connections: The input and output state bits are not connected. You will need to wire them in *magic*. You may wish to use the wiring tool in *magic* to do this. It is recommended to do the wiring in *metal*1 using *polycontact* to connect to the *poly* stubs that *mpla* provides. It is important to add the feedback connections within the *magic* cell that contains the PLA. If the feedback connections are added in a higher level cell, then there will be extractor warnings. It greatly helps the design process to prevent extractor warnings.

RESTART Signal: Always include a **RESTART** signal in your *meg* description of the PLA. It improves the testability of your chip to connect this **RESTART** line directly to an I/O pad. The **RESTART** signal should cause the PLA to go to *state*0 and therefore make the internals of your project more observable. Your *irsim* test vectors should always begin with a **RESTART** on initialization to clear the undefined state bits.

State Bits: It is also a good idea to bring each of the state bits out to an I/O pad if space permits in your projects. This greatly improves the testability of your design. You should watch the state bits through IRSIM to observe if the PLA is making the appropriate transitions.

Meg example for counter

The following counter has **RESTART** and counts 0 - 3 - 1 - 2. The *meg* program is shown below and on the following pages is the output of *mpla*. Note that "–" begins a comment.

```
--      MEG program
--      2-bit counter,  0,3,1,2
--      with even/odd output
--      state transition on every clock

INPUTS:  RESTART;
OUTPUTS: ODD;

RESET ON RESTART TO state0;

state0  :       GOTO state3 (ODD=1);
state1  :       GOTO state2 (ODD=0);
state2  :       GOTO state0 (ODD=0);
state3  :       GOTO state1 (ODD=1);
```

Meg.summary File

It is useful to review the output file *meg.summary* which gives a listing of the Inputs, Outputs, and State Table. Note that $StBit0*$ is the most significant bit of the state encoding, and that $StBit1*$ is the least significant bit. Also note that the order of the state bits in the "INPUTS" and "OUTPUTS" list is swapped to correspond with the actual *magic* layout. The reason for this is to simplify the final connection of the state bits, by eliminating the need for bridging to a different routing layer.

For simple counters, you can use the state bits as the counter output. The *meg* program usually assigns the binary encoding of the states in a first come, first served manner. Therefore in the input *meg* file, I intentionally listed the states in the counting order that I wanted.

```
SUMMARY INFORMATION GENERATED BY MEG FROM FILE counter.meg
meg:  Number of inputs  =1
meg:  Number of outputs =1
meg:  Number of states  =4
meg:  State field size  =2 * 2
meg:  Table width       =6


INPUTS:
   i00:  RESTART
   s00:  StBit0* (msb)
   s01:  StBit1* (lsb)


OUTPUTS:
   n01:  StBit1* (lsb)
   n00:  StBit0* (msb)
   o00:  ODD


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

     0  0  0  1  1  1   state0

     0  0  1  0  1  0   state1

     0  1  0  0  0  0   state2

     0  1  1  1  0  1   state3
```

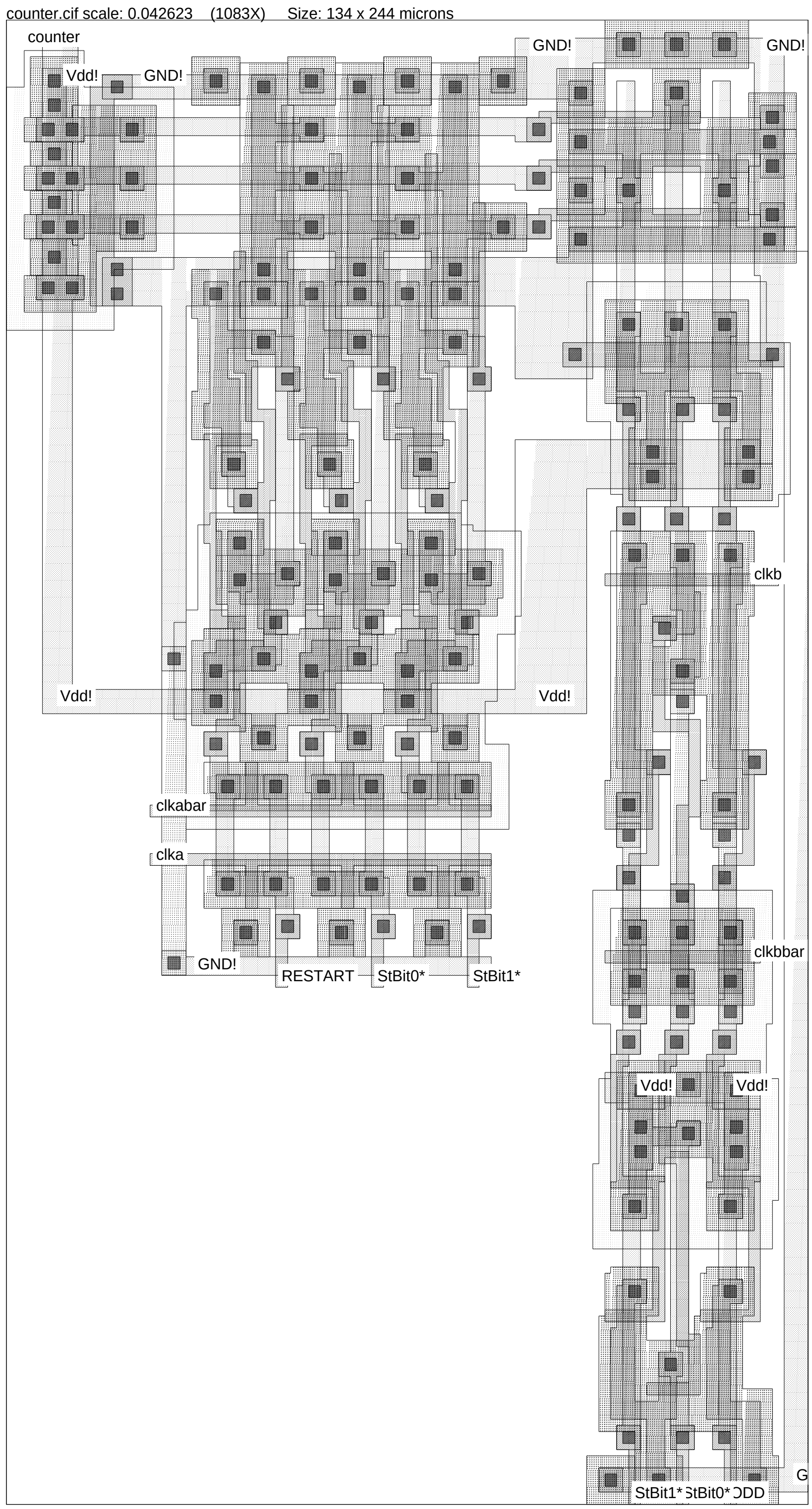

Figure 1: Magic layout from *mpla* before Vdd and state bit connections.

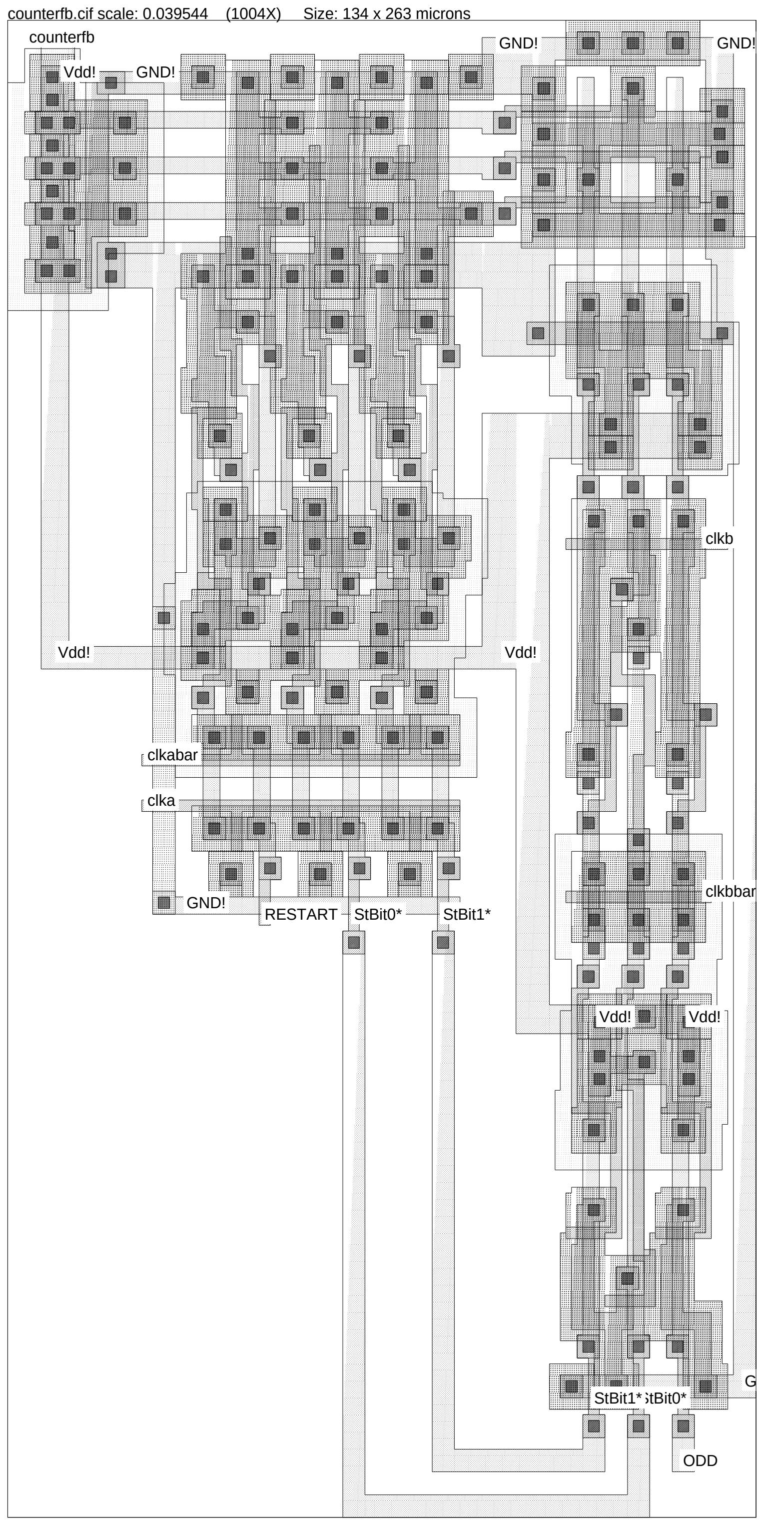

Figure 2: Magic layout after connection of Vdd and state bits.

Counter Simulation with IRSIM

The behavior of the counter circuit can be verified with IRSIM. A sample command file should first reset the PLA and then observe operation for several cycles. Since this is a simple counter, it begins to count immediately after reset.

```
| 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 StBit0* StBit1*
ana StBit0* StBit1* STATE RESTART ODD
V RESTART 0 1 0 0 0 0 0 0 1 0 0 0 0
R
```

Both two-phase clocks and their complements are defined in "clock" commands. In an actual circuit, you would need to provide inverter cells to generate $CLKABAR$ and $CLKBBAR$. A vector, $STATE$, has also been defined to observe the state bits. It was ordered with the msb first, so that an easy to recognize octal value could be displayed in IRSIM. The stimulus vector "V" is defined to RESTART the PLA. The RESTART intentionally occurred on the second clock cycle so that it can be observed that the PLA powers-up in an undefined state. The PLA was allowed to operate for six clock cycles, and then the RESTART signal was asserted again to show that the PLA would return to the initial state.

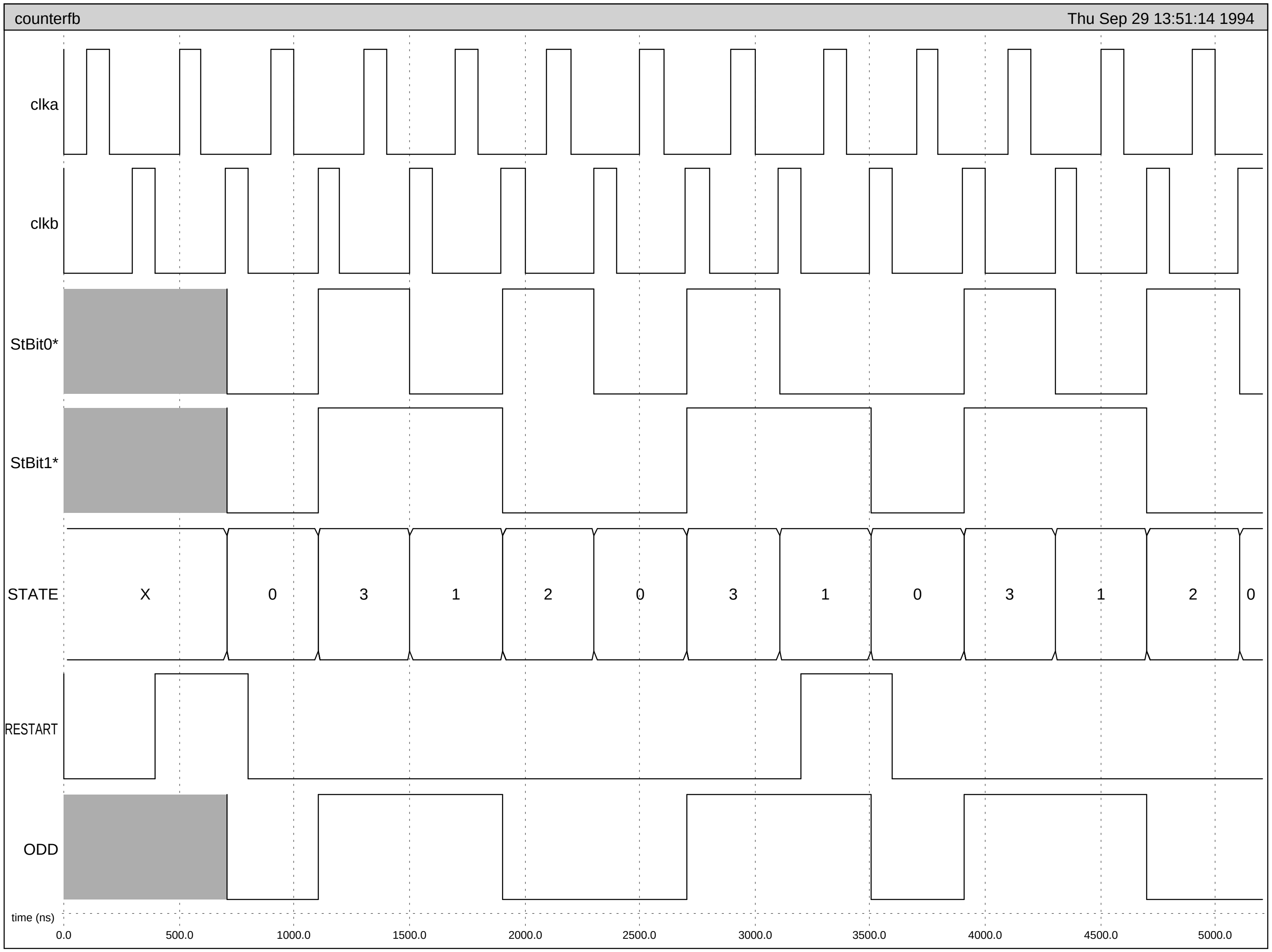

Figure 3: IRSIM simulation of counter PLA.