تصميم الدارات الإلكترونية بالحاسوب

Computer Design of Electronic Circuits

المحاضرة التاسعة

السنة الرابعة قسم التحكم والأتمتة

العام الدراسي 2023-2024

محتويات المقرر

1. مدخل إلى أهمية تطوير أدوات التصميم باستخدام الحاسب (Introduction to The Need of Developing CAD Tools)
2. تصنيف عام لأنواع أدوات التصميم (General Classification of CAD Tools Used in Electronic Systems Design)
3. مدخل إلى اللغات المستخدمة في التصميم (Introduction to Design Languages VHDL, Verilog, Verilog System, ..)
4. مدخل إلى مراحل بناء النظم الرقمية (Introduction to Digital Systems Synthesis)
5. مرحلة البناء منخفض المستوى (Low Level Synthesis)
6. تصميم الدارات المتكاملة للنظم عالية التكامل (Layout Design for VLSI Systems)
7. تطبيقات تصميمية (Design Applications)
8. اتجاهات التطور الحديثة (Trends and New Directions)

Verilog code for up counter


```
module up_counter(input clk, reset, output[3:0] counter);
reg [3:0] counter_up;
// up counter
always @(posedge clk or posedge reset)
begin
if(reset)
 counter_up <= 4'b0;
else
 counter_up <= counter_up + 4'b0001;
end
assign counter = counter_up;
endmodule
```

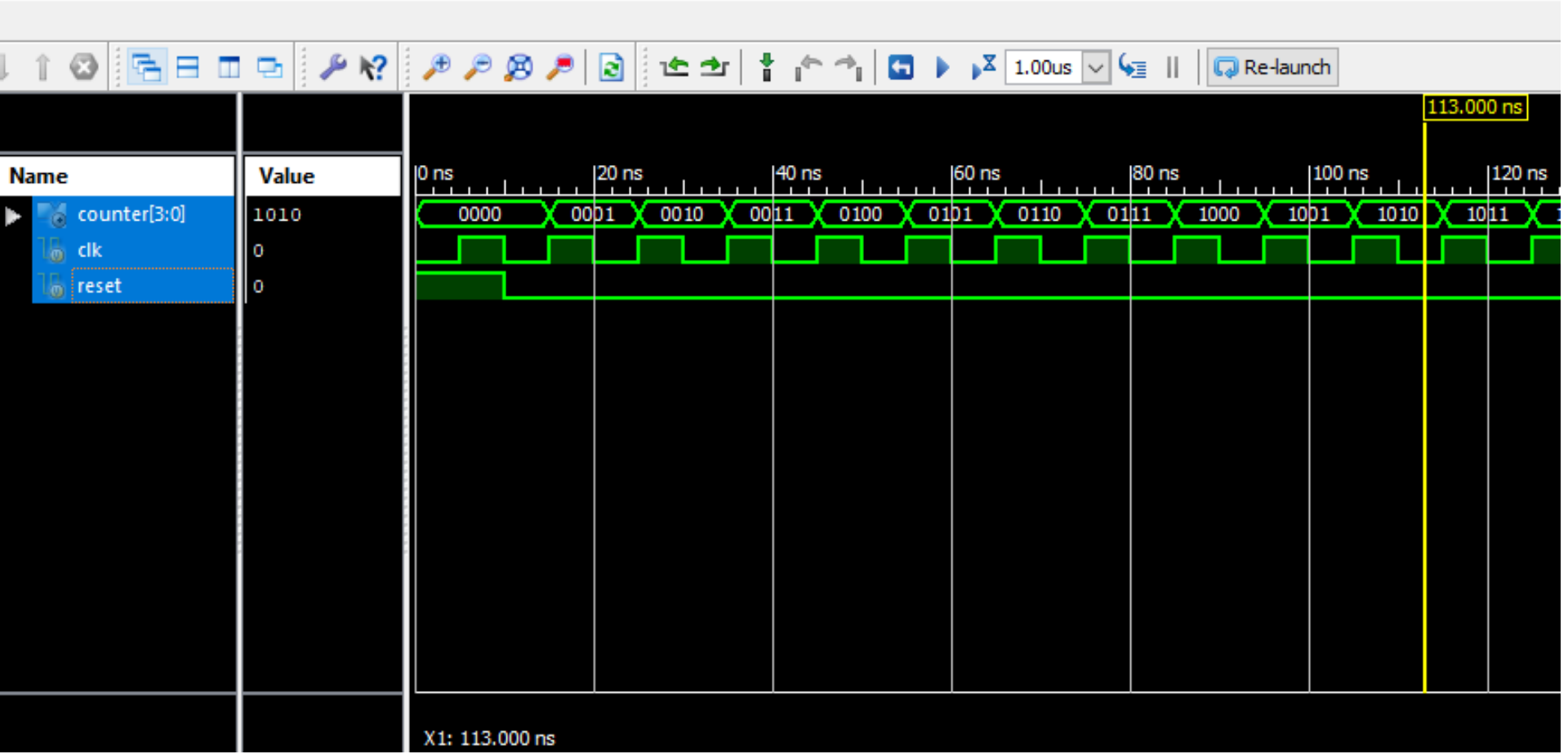
Verilog code for up counter test

```
module upcounter_testbench;
    // Inputs
    reg clk;
    reg reset;
    // Outputs
    wire [3:0] counter;
    // Instantiate the Unit Under Test (UUT)
    up_counter uut (
        .clk(clk),
        .reset(reset),
        .counter(counter) );
always #5 clk=~clk;
    initial begin
        // Initialize Inputs
clk = 0;
reset=1;
#10;
reset=0;
#10;
end
endmodule
```

Or

up_counter uut (clk, reset, counter)

Verilog code for up counter test



// Example

Clock Divider

Clock divide frequencies

q[i]	Frequency (Hz)	Period (ms)
i	50000000.00	0.00002
0	25000000.00	0.00004
1	12500000.00	0.00008
2	6250000.00	0.00016
3	3125000.00	0.00032
4	1562500.00	0.00064
5	781250.00	0.00128
6	390625.00	0.00256
7	195312.50	0.00512
8	97656.25	0.01024
9	48828.13	0.02048
10	24414.06	0.04096
11	12207.03	0.08192
12	6103.52	0.16384
13	3051.76	0.32768
14	1525.88	0.65536
15	762.94	1.31072
16	381.47	2.62144
17	190.73	5.24288
18	95.37	10.48576
19	47.68	20.97152
20	23.84	41.94304
21	11.92	83.88608
22	5.96	167.77216
23	2.98	335.54432

```
module clkdiv (
input clk ,
input clr ,
output clk190 ,
output clk25 ,
output clk3 );
reg [23:0] q;
// 24-bit counter
always @(posedge clk or posedge clr)
begin
if(clr == 1) q <= 0;
else q <= q + 1;
end
assign clk190 = q[17]; // 190 Hz
assign clk25 = q[0]; // 25 MHz
assign clk3 = q[23]; // 3 Hz
endmodule
```

Digital tools

OpenLane - end to end ASIC flow
OpenROAD - provides many of the tools in OpenLane
Silicon Compiler - end to end ASIC flow
Coriolis 2 - end to end ASIC flow
OSS Cad Suite - lots of open source tools useful for digital design
OSFPGA - end to end FPGA flow with open source tools such as Yosys, VTR and VPR
VHDL support - with GHDL
Awesome list of verification tools
Awesome list of HDL tools / libraries / cores ...
SemiWiki's list of open EDA tools
Andreas' list of awesome hardware tools

Analog tools

Klayout - modern style layout drawing tool.
Xschem - old school, schematic capture
Mosaic - schematic capture (experimental)
Ngspice - simulation
Xyce - simulation
gdsfactory - EDA tool to Layout and simulate Integrated Circuits.

Generators

OpenRAM - SRAM generator
DFFRAM - Memory Compiler using FF/Latch cells
OpenFASoC - Analogue IP generator (LDO, temperature sense etc)
MOSAIC_BAG2 - Analogue IP generator framework (w/minimum working example)
RgGen - CSR generator (SystemVerilog/Verilog/VHDL RTL, UVM reg model etc)

Companies

- ChipFlow - Helping product companies to make their own chips with open source tools
- Efabless - simplifying chip creation
- LibreSilicon - open source manufacturing process standard
- LowRISC - open source silicon designs and tools
- Skywater Technology - foundry that first published open source PDK
- OpenHW Group - open-source cores, verification, software and standards
- YosysHQ - open source EDA & Formal Verification
- ZeroASIC - makers of silicon compiler ASIC tool flow

ChipFlow

Product About Docs Blog

Build custom ICs with your existing team!

Our unique Python flow for microchip design enables your team in designing the right silicon for your product at a fraction of cost and complexity - from inception to manufacturing.

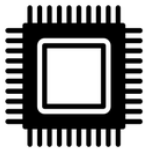
مثال

Code

Python based hardware design

Mask

Automated layout using end-to-end integration



Chip

API driven manufacturing, optimised for lower volumes

Ship

API driven fulfilment

I believe that is the future in the next decade

الأسئلة والمناقشة

Dr. Aula Jazmati