



Advanced Clock Environment (ACE)

Overview

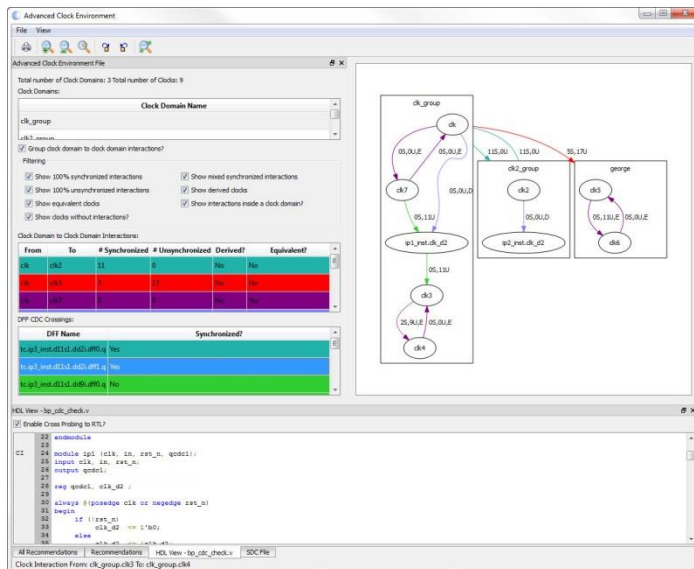
The Blue Pearl Software's Visual Verification Suite is the next generation EDA solution for ASIC, IP and FPGA verification that automates RTL analysis, CDC analysis, and SDC generation.

The Suite augments existing EDA and FPGA vendor tool flows with a native Windows or Linux user experience.

The Advanced Clock Environment (ACE) tool solves the iterative and reactive CDC setup problem experienced by designers. It is used before running a CDC analysis. With ACE, designers can clearly see if clocks are not in the intended domains and make corrections before in-depth CDC analysis.

What is ACE?

Blue Pearl Software's Advanced Clock Environment provides a graphical representation summarizing data paths between clocks and can make recommendations for grouping of clocks into clock domains. With ACE, designers can identify clocks to better understand how they interact with synchronizers in the design. This allows users to quickly identify improper synchronizers or clock domain groupings that cause CDC metastability.



From	To	# Synchronized	# Unsynchronized	Derived?	Equivalent?
clk	clk2	11	0	No	No
clk	clk3	11	0	No	No
clk	clk4	11	0	No	No

```

22 endmodule
23
24 module sp1 (clk_in, rst_n, qout1);
25 input clk_in, rst_n;
26 output qout1;
27
28 reg qout1, clk_div2;
29
30 always @(posedge clk_in or negedge rst_n)
31 begin
32     if (!rst_n)
33         clk_div2 <= 1'b0;
34     else
35         clk_div2 <= clk_in >> 1;
36     end
37 end
  
```

Why should I be concerned with metastability and CDC?

At its most basic level, metastability is what happens within a register when data changes too soon before or after the active clock edge; that is, when setup or hold times are violated. A register in a metastable state is in between valid logic states, and the process of settling to a valid logic state takes much longer than normal. It will eventually fall into a stable "1" or "0" state, but there is no way to predict which way it will fall or how long it will take. If the metastable state lasts longer than one clock cycle, it can be transferred to the next register.

When data is transferred between two registers whose clocks are asynchronous, metastability will happen. There is no way to prevent it. All you can do is to minimize its impact by placing the two clocks in different clock domains and using a clock synchronization technique at the crossing point. Hence the name "clock domain crossing" (CDC).

Putting two clocks into the same clock domain is a declaration that these two clocks are synchronous to each other, and CDCs between them do not need to be synchronized. If the clocks are from the same source, or one is derived from the other, then they are synchronous and should be placed into the same clock domain.

Clocks that are asynchronous to one another should always be placed in different clock domains, and any CDCs between them should be synchronized. Even two clocks of the same frequency should be placed into different domains if they come from independent sources. All specifications have tolerances or "error bars," and two independent clock sources of the same frequency will drift relative to one another over time. Failing to synchronize CDCs between two such clocks will cause metastability problems.

What can I do with ACE?

- Evaluate clock domain definitions
- Evaluate SDC clock constraints
- Generate a graphical analysis of clock and clock domains
- Validate clock grouping recommendations
- Generate SDC template to be used by a CDC analysis tool

Features of ACE

- Detailed analysis of clock, clock groupings, and interactions
- Visual display filters enable quick debug
- Pinpoint exact location of problem in RTL design

Advanced Clock Environment

Advanced Clock Environment File

Total number of Clock Domains: 2 Total number of Clkds: 9

Clock Domains:

dk1_group

Group clock domain to clock domain interactions?

Filtering

Show 100% synchronized interactions Show mixed synchronized interactions

Show 100% unsynchronized interactions Show derived clock

Show equivalent clock Show interactions inside a clock domain?

Show clock without interactions?

Clock Domain to Clock Domain Interactions:

From	To	# Synchronized	# Unsynchronized	Desired?	Equivalent?
dk1	dk2	11	0	No	No
dk2	dk1	0	11	No	No
dk1	dk3	0	0	No	No
dk3	dk1	0	0	No	No
dk1	dk4	0	0	No	No
dk4	dk1	0	0	No	No
dk1	dk5	0	0	No	No
dk5	dk1	0	0	No	No
dk1	dk6	0	0	No	No
dk6	dk1	0	0	No	No
dk1	dk7	0	0	No	No
dk7	dk1	0	0	No	No
dk1	dk8	0	0	No	No
dk8	dk1	0	0	No	No
dk1	dk9	0	0	No	No
dk9	dk1	0	0	No	No
dk1	dk10	0	0	No	No
dk10	dk1	0	0	No	No
dk1	dk11	0	0	No	No
dk11	dk1	0	0	No	No
dk1	dk12	0	0	No	No
dk12	dk1	0	0	No	No
dk1	dk13	0	0	No	No
dk13	dk1	0	0	No	No
dk1	dk14	0	0	No	No
dk14	dk1	0	0	No	No
dk1	dk15	0	0	No	No
dk15	dk1	0	0	No	No
dk1	dk16	0	0	No	No
dk16	dk1	0	0	No	No
dk1	dk17	0	0	No	No
dk17	dk1	0	0	No	No
dk1	dk18	0	0	No	No
dk18	dk1	0	0	No	No
dk1	dk19	0	0	No	No
dk19	dk1	0	0	No	No
dk1	dk20	0	0	No	No
dk20	dk1	0	0	No	No
dk1	dk21	0	0	No	No
dk21	dk1	0	0	No	No
dk1	dk22	0	0	No	No
dk22	dk1	0	0	No	No
dk1	dk23	0	0	No	No
dk23	dk1	0	0	No	No
dk1	dk24	0	0	No	No
dk24	dk1	0	0	No	No
dk1	dk25	0	0	No	No
dk25	dk1	0	0	No	No
dk1	dk26	0	0	No	No
dk26	dk1	0	0	No	No
dk1	dk27	0	0	No	No
dk27	dk1	0	0	No	No
dk1	dk28	0	0	No	No
dk28	dk1	0	0	No	No
dk1	dk29	0	0	No	No
dk29	dk1	0	0	No	No
dk1	dk30	0	0	No	No
dk30	dk1	0	0	No	No
dk1	dk31	0	0	No	No
dk31	dk1	0	0	No	No
dk1	dk32	0	0	No	No
dk32	dk1	0	0	No	No
dk1	dk33	0	0	No	No
dk33	dk1	0	0	No	No
dk1	dk34	0	0	No	No
dk34	dk1	0	0	No	No
dk1	dk35	0	0	No	No
dk35	dk1	0	0	No	No
dk1	dk36	0	0	No	No
dk36	dk1	0	0	No	No
dk1	dk37	0	0	No	No
dk37	dk1	0	0	No	No
dk1	dk38	0	0	No	No
dk38	dk1	0	0	No	No
dk1	dk39	0	0	No	No
dk39	dk1	0	0	No	No
dk1	dk40	0	0	No	No
dk40	dk1	0	0	No	No
dk1	dk41	0	0	No	No
dk41	dk1	0	0	No	No
dk1	dk42	0	0	No	No
dk42	dk1	0	0	No	No
dk1	dk43	0	0	No	No
dk43	dk1	0	0	No	No
dk1	dk44	0	0	No	No
dk44	dk1	0	0	No	No
dk1	dk45	0	0	No	No
dk45	dk1	0	0	No	No
dk1	dk46	0	0	No	No
dk46	dk1	0	0	No	No
dk1	dk47	0	0	No	No
dk47	dk1	0	0	No	No
dk1	dk48	0	0	No	No
dk48	dk1	0	0	No	No
dk1	dk49	0	0	No	No
dk49	dk1	0	0	No	No
dk1	dk50	0	0	No	No
dk50	dk1	0	0	No	No

DFT CDC Coverage:

DFT Name	Synchronized?
icp1_int4111-460-090.g	Yes
icp1_int4111-460-091.g	Yes
icp1_int4111-460-090.g	No

HDL View - bp_cdc_check.v

Enable Cross Probing to RTL?

```
28 module top (clk, in, rst_n, qout);
29 input clk, in, rst_n;
30 output qout;
31
32 reg qout;
33
34 always @(posedge clk or negedge rst_n)
35 begin
36     if (rst_n)
37         qout <= 1'b0;
38     else
39         qout <= in;
40 end
41 endmodule
```

All Recommendations | Recommendations | HDL View - bp_cdc_check.v | SDC File

Clock Interaction From: dk1_group To: dk2_group

Why should I use ACE?

- Without proper clock domain setup, CDC analysis results are unreliable.
- Virtually impossible for any tool to automatically deduce all the clock domains
- Designers perform many CDC iterations as they incrementally make clock / clock domain decisions

The overall goal of Advanced Clock Environment is to enable engineers to find metastability issues in designs by properly grouping clocks into clock domains. Design and Verification engineers use ACE to ensure the clock domains are properly specified before running a CDC analysis.

ACE will quickly find errors in clock domain groupings or find/recommend appropriate clock domain groupings for a circuit that is synchronized.

Engineers who are interested in reducing cycle times, by identifying problems early on, find Blue Pearl's Advanced Clock Environment a great addition to their existing design environment.

Contact us:

www.bluepearlsoftware.com

+1(408) 961 0121

sales@bluepearlsoftware.com