Pakistan Association of Anthropology, Islamabad, Pakistan
Special issue
Sci.int.(Lahore),27(2),1671-1674,2015 ISSN 1013-5316; CODEN: SINTE 8

EFFICIENT RESET DESIGN TECHNIQUES

Zoha Qamar[1], Bushra Yaseen[2], Usman Maqbool[3]
[1]Lecturer, Govt. Degree College for Women, Peshawar Road Rawalpindi,
[2]Department of Electrical Engineering, COMSATS Islamabad,
[3]Department of Electronics, Quaid-i-Azam University, Islamabad
Corresponding Author's Email: zoha.qamer@gmail.com

ABSTRACT: *Efficient Reset design techniques for the digital devices like ASIC, FPGA, CPLD etc can be tricky. Reset architecture is the general culprits of the Meta stability behavior in any digital design. The paper will study the advantages and disadvantages of synchronous and asynchronous resets design. Then compare the each type of reset and present local auto asynchronous reset architecture that can process up to max latency and throughput of the design. The results were compared with the available literature and the performance was found very encouraging.*

Key Words: Reset Design,

INTRODUCTION

Reset designs are composite and not often this subject is concentrated in engineering schools [1]. In Literature the topic of reset design have two kind of resets. Asynchronous reset and synchronous reset. Asynchronous signal have top priority to any other signals in the design. Asynchronous resets do not require clocks. They have separate path and do not share the path with any other signal. Either to use synchronous or asynchronous resets is the biggest problem in digital designs.

This is the obvious advantage, designs which are using synchronous are 100% synchronous with clock, satisfying all the setup and hold time requirement [2]. But sometime to have the fully synchronous circuits every time may not be mandatory in such cases asynchronous reset are preferable.

Synchronous reset architecture flexibility, with which the unpredictable behavior of design can be avoided, makes FPGAs an obvious choice as the test bed for the proposed hardware [3]. Obviously, there are significant pros and cons for using each reset type and their impact can be very effective in the design performance. Selecting a reset type is a very important matter in the design. This paper models the digital design circuits in Verilog-2001 ANSI coding style.

The paper is organized as follows. Section 2 discusses the reset purpose and its background work. Section 3 & 4 describes pros & cons of synchronous and asynchronous reset. The area that forms ground for our proposed case study algorithms presented in proposed hardware Section 5.

Results and their analysis are given in Section 6 covers comparison & discussion, while section 7 concludes the suggest future work.

Reset Purpose

Computer system comes with hardware reset which put back the system into a known working state (restart state) by pressing the restart button [4]. Resets are mandatory for the digital design to let the system remain in known state.

Global Reset Problem

The core of all digital architecture is flip-flop, they are the single bit register, use to hold the data between the two consecutive edge of the clock .They actually do not have reset architecture in them built-in. Reset logic is provided to flip-flop to reduce the data logic path greatly. Applying reset to the digital circuits allow logic to go into known state. Hidden glitches (unwanted signals) are always present in asynchronous design, which forced flip-flops to be in the unknown state.

Global reset in the digital design is not recommended generally because of the extensive amount of routing resources used in design. Logic resources are widely used in design where increases in size minimize the design efficiency and require higher speed grade.

Before Selecting Global reset methodology for FPGA design few things must be considered, what to choose asynchronous or synchronous, how each flip-flop will get a reset, and how reset tree will be buffer and the verification of its timing. How to apply test scan vector on the reset design and to check the functionality of different logics of design [5].

Synchronous Reset

Synchronous resets will only have an effect on the flip-flop on the active edge of a clock. And they function as part of the combinational logic which generates the d-input to the flip-flop [6]. To model the design for coding purpose, reset would be in the if part of if/else style and all the combinational logic in else portion. The problem could arise if the coding style is not followed properly, for hardware it would create glitches in design and for simulator the design would limit the reset to reach at each flip flop. The core objectives of a reset are to put the design into a known state for both synthesis and simulation. High fan out of a reset tree makes reset as a delayed signal in comparison to clock signal.

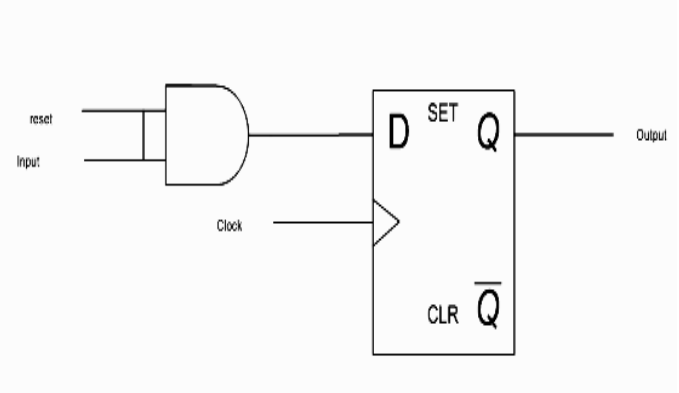

Fig. 1: Schematic of Synchronous Reset

Module synchronous_reset_design (output reg out,
Input in, clock, reset); always @ (posedge clock)

if (!reset) out <= 1'b0;else out <= in;endmodule.

Example. 1: Synchronous reset design code in Verilog-2001 style

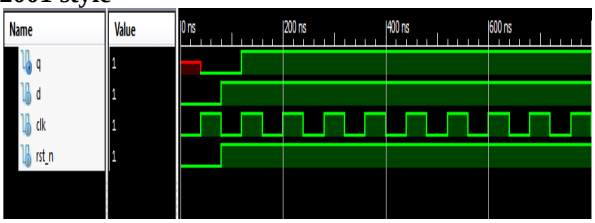

Fig2. Post layout Simulation of Synchronous reset

Advantages of synchronous resets

For the efficient design technique synchronous reset has to be applied on the active clock edge .It safe the design from the hidden glitches. But it does not provide complete protection from accidental glitches which can occur near the positive edge of the clock and satisfy both setup and hold time conditions for making the flip-flops to be reset [7]. If reset design is produced from the combinational logic then chances of having random glitches gets higher.

Disadvantages of synchronous resets

Designs which are not area efficient requires large routing resources to satisfy the timing constraints .Synchronous reset has to meet the setup and hold time of the clock , for that reset pulse has to be wide enough on every active edge of the clock on the reset distribution tree.

Asynchronous Reset

In asynchronous reset design a reset input is implied into the flip-flop design directly [8]. Usually the reset pulse is active low and the flip-flops attach with that reset pins goes with low level logic. Due to the inappropriate coding of asynchronous resets in digital logic design are the reasons of design failures. The prime problem with asynchronous resets is reset removal which is also called as reset de-assertion.

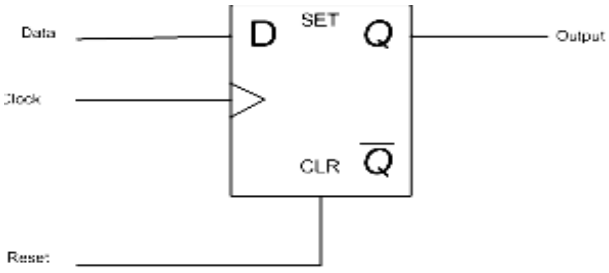

Fig. 3: Schematic of Asynchronous reset

Module asynchronous_reset_design (output reg out, input in, clock, reset); always @ (posedge clock or negedge reset)if (!reset) out <= 1'b0; else out <= in;endmodule.

Example. 2: Asynchronous reset design code in Verilog-2001 style

Advantage of Asynchronous Reset

Irrespective of the positive clock edge Flip-flop instantaneously takes the value of reset. No clock

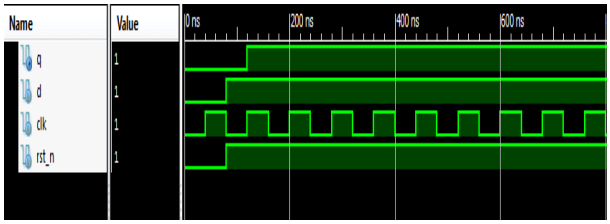

Fig3: Post layout Simulation of A synchronous reset

synchronization is required. Asynchronous input reset signal (like push button reset) provides Faster data path in design. Asynchronous reset designs reduce logic resources by removing AND gate at the input of the flop, which reduces the complexity of routing resources [9]. This could be double edged sword; it would be an advantage if design allows the use of asynchronous reset.

Disadvantage of Asynchronous Reset

Asynchronous reset have major problem with its reset release time. To the designers Surprise we consider the reset for its time of assertion only. The release of the reset in the system near the active edge of clock (non-controlled environment) can be a reason for the flip-flop to be in unknown state, thus making the violations in design . Design of the asynchronous reset is implemented via NAND gate feedback loop. When the NAND gate input goes low level and makes output to be low irrespective of the feedback loop of input. Except when desertion reset state comes, that NAND gate act as an inverter and flip-fop gets in normal state, which is vulnerable for the setup and hold time conditions. Synchronous reset don't have this problem as the design is completely synchronous with clock rating which satisfy both setup and hold time requirement on reset as well as on input .Both reset and data are AND-ed and fed as an input to the flip-flop. Asynchronous resets cause modules to come out of reset at different times due to inconsistent path delays

Proposed Hardware – Local Auto Asynchronous Reset

The local auto reset synchronizer logic is designed to have the best pros of both rest types. An internal reset signal is created which asynchronously resets first flip-flops, which buffer the reset tree for the rest of the flip flops in the design. So that whole design gets reset asynchronously. Reset release is done by removing the reset signal via reset synchronizer, which buffers the input of the first master reset flip-flop (which is active high). Synchronizer design usually takes two clock cycles for the reset release of master reset with the two flip-flops .First flip-flop is required to synchronies the reset pulse with positive edge of clock where the second flip-flop remove any meta stability which can be caused from the reset signal being de-asserted near the active edge of clock .

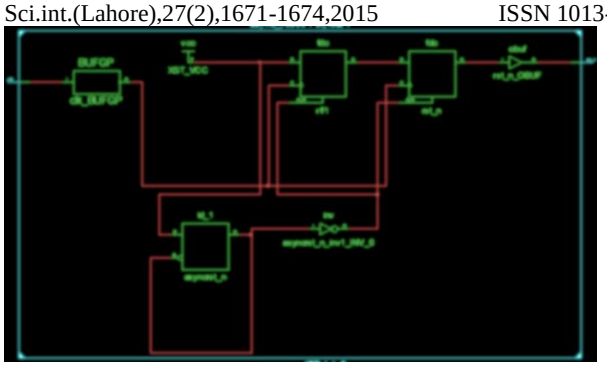

Fig. 4 Technology Schematic of local-auto reset circuit created by the synthesizer

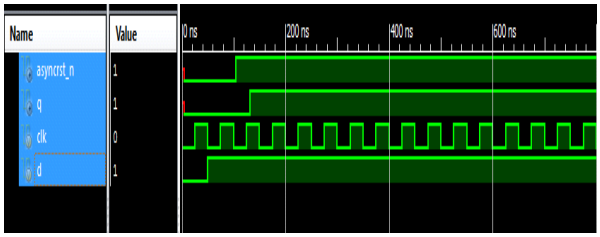

Fig.5: Post layout Simulation of Local Auto asynchronous reset

Table. 1: Comparison of Frequencies and Logic Resources with Three Reset Circuits on two Different FPGA Technologies

Devices	Generated Frequency MHz			Resource Utilization								
	Proposed Reset	Asy-Reset	Syn-Reset	Proposed Reset			Asyn-Reset			Syn-Reset		
				Luts	Flip-Flop	Slices	Luts	Flip-Flop	Slices	Luts	Flip-flop	Slices
xc3s1500-4fg320	230	230	230	22	10	11	21	9	10	21	9	10
Xc3s200-5ft256	267	267	267	22	10	11	21	9	10	21	9	10

In reset synchronizer the first flip-flop may have hidden meta stability issue since its input is set to high level logic, the output has been reset to a logic 0 asynchronously in the specific reset removal time of the flip-flop (the reset goes high near the positive edge of the clock input to the same flip-flop). So second flip-flop is mandatory. The second flip-flop of the reset synchronizer is not responsible for removal time Meta stability because the input and output of the flip-flop have value logic 0 when reset is removed. There is no logical difference between the input and output of the flip-flop so it is not possible for output to oscillate between two different logic values.

CONCLUSIONS

Inserting a global reset will affect the timing constraints, area constraints .With the trend towards higher speed clocks and small size design, the consistency is a prime objective. Reset of the particular combinational logic must be identified and the assertion of reset pin with wide pulse should be carefully controlled with in a synchronous circuit. Designs with small no. of resources tend to have higher performance by potentially saving a speed grade of the device.

Properly used, synchronous and asynchronous resets in digital design are really efficient for the reliable reset assertion. Even though an asynchronous reset is a best method to reset the design. Release of an asynchronous reset pin creates major issues if they are not modeled correctly. The appropriate method is to model asynchronous resets with the auto reset synchronizer logic. Which allow asynchronous reset in design and assure synchronous reset removal with normal design functionality with little pay-off in logic resources in comparison of synchronous and asynchronous reset design? Either synchronous or asynchronous resets design , using auto synchronizer as described in this paper can be worth notable for designer since the issues related to routing, buffering and timing can be sort out.

REFERENCES

[1] IEEE Standard Verilog Hardware Description Language, IEEE Computer Society, IEEE, *New York, NY, IEEE Std* 1364 (2001)

[2] Keating, M. and Bricaud, P. "*Reuse Methodology Manual*", Second Edition, Kluwer Academic Publishers. pp 35 (1999)

[3]Synopsys SolvNet, Doc Name: 903391, "Methodology and limitations of synthesis for synchronous set and reset," Updated 09/07/2001 - solvnet.synopsys.com/retrieve/903391.html

[4] Synopsys SolvNet, Doc Name: 902298, "Recovery and Removal timing checks on Primetime," Updated02/13/2002solvnet.synopsys.com/retrieve/902298.html

[5]Cummings, C. E. "Nonblocking Assignments in Verilog Synthesis, Coding Styles That Kill!," *SNUG (Synopsys Users Group) 2000 User Papers*, section-MC1 (1 st paper) (2000)

[6]Cummings, C.E. and Mills, D. "Synchronous Resets? Asynchronous Resets? I am so confused! How will I ever know which to use?" SNUG (Synopsys Users Group) San Jose, 2002 User Papers (2002)

[7] Mills, D and Cummings, C. E. "RTL Coding Styles That Yield Simulation and Synthesis Mismatches," SNUG *(Synopsys Users Group) Proceedings, section-TA2 (2ndpaper), March* (1999)

[8]ESNUG #60, Item 1- www.deepchip.com/items/0060-01.html

[9] ESNUG #240, Item 7- www.deepchip.com/items/0240-07.html