ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Design of Thruster Driver Module using Synchronous Design Technique in VHDL

Priyanka P Desai[1], Deepa Pramod[2], Dushyanth N D[3]

M. Tech Student, Department of Electronics and Communication, Jain University, Bengaluru, India [1]

Engineer SE, Control and Digital Electronics Group (CDEG), ISRO Satellite Centre, Bengaluru, India[2]

Asst. Professor, Department of Electronics and Communication, Jain University, Bengaluru, India [3]

Abstract: A thruster is a propulsive device or an actuator used in the spacecraft. The primary function of the thrusters is to control the attitude and orbit of the satellite. This paper presents the design of thruster driver module using synchronous design technique in VHDL. A 1 MHz common clock is used to synchronize the design interfaces of the thruster driver module. By implementing this new synchronous design technique, all the signals are synchronized and sampled at a well-defined clock interval. The simulation results show, the synchronous design technique implemented is more reliable, efficient and can be best used for testing when compared to the existing asynchronous method. The synthesis results show the number of core cells is being reduced, thus minimizing the area when compared to the existing asynchronous design.

Keywords: Thruster Driver Module, Synchronous Design, Common Clock, Interface Designs, Trigger Pulses.

I. INTRODUCTION

A spacecraft plays an important role in today's modern world. It provides inter-connectivity to the world. A satellite consists of two major parts- payloads and mainframes. The payload provides service to the satellite users and mainframes provide major platforms for its payload [5].

The mainframe of the satellite consists of the subsystems such as telemetry, telecommand, power, attitude and orbit control systems (AOCS). Thrusters are the actuators present in the AOCS which provide interface to the attitude and orbit control electronics (AOCE) [3]. These thrusters are mounted on the satellite to generate required thrust to maintain the required attitude of the satellite.

The existing thruster driver module is based on asynchronous design techniques. The challenges faced by asynchronous design technique are as follows [2]: (1) Testability and determining the performance of the circuit with asynchronous clocking events is difficult. (2) They require more number of clocks. (3) Area required is more. (4) They are difficult to design. (5) They often require more transition on the computation path. (6) Removal of hazards is more cumbersome. (7) Asynchronous circuits generally require extra time due to signaling policies, thus increasing average case delay.

The above mentioned drawbacks can be avoided by converting an asynchronous design to a synchronous design. Synchronous designs are more reliable and deterministic in behavior due to the fact that all signals are sampled at well-defined time interval.

Synchronous design rely upon few parameters to guarantee the operation, namely, the maximum frequency (fmax) , register and setup times (tSU and tH), clock to output (tCO) time [2]. They can be tested easily and run statically.

This paper presents the design of thruster driver module using synchronous design technique in VHDL. It consists of the interface designs of thruster driver interface, serial digital channel interface, internal command decoding and thruster history monitoring used in thrusters of the spacecraft. These interface designs are synchronized with 1 MHz common clock.

The paper is organized as follows. Overview of thruster driver module is discussed in section II. The proposed methodology of synchronization is discussed in section III. Section IV discusses the interface designs of thruster driver module which includes the synchronous design of proposed work, i.e., the IO decoder is discussed in sub section A, command generation is discussed in sub section B, serial digital channel interface is discussed in sub section C, thruster driver interface is discussed in sub section D, subsection E discusses about thruster history monitoring, data ready interface and clock and mode generation are discussed in sub section F and G respectively. The simulation results are discussed in section V. Section VI s

II. THRUSTER DRIVER MODULE

The thruster driver module has 87 functional inputs and 96 functional output lines [4]. The main function of this module is to provide

- Thruster driver interface
- Thruster history monitoring and
- Internal command decoding

Table I shows the specification and the mapping of 87 functional input and 96 functional outputs of the thruster driver (TD) module.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

TABLE I: TD MODULE SPECIFICATIONS

SL.No.	Specifications	Capability
1.	Thruster driver interface	16
2.	Serial digital channel interface	18
3.	Spare input line Interface	36
4.	Spare output line Interface	59

The module inputs ip1 (86:0) and module outputs out1 (95:0) are routed through router module.

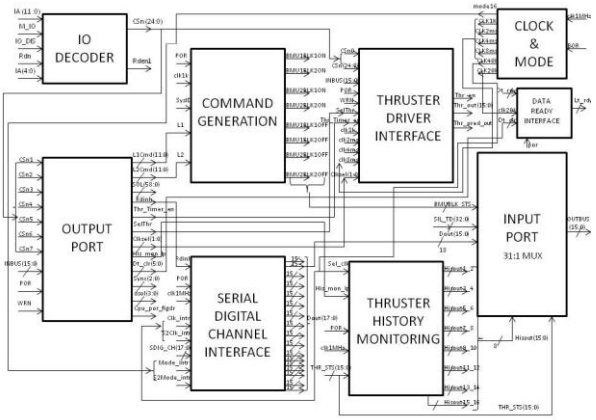

Fig. 1 Block diagram for Thruster Driver Module

Fig. 1 shows the block diagram of thruster driver module. It consists of the interface design of IO decoder, command generation, serial digital channel interface, thruster driver interface, thruster history monitoring clock and mode generation, data ready interface and input-output ports. A 1 MHz global clock is used for the synchronization of the entire TD module. The input port is used to route the output signals of the TD Module to the other subsystems of the spacecrafts. The output port routs the data from the processors

III. PROPOSED SYNCHRONOUS DESIGN TECHNIQUE

The AOCE of a spacecraft is a real time system. Real time systems are interrupt-driven i.e., the design interfaces receives different trigger pulses from the microprocessor which are given directly as an input to the clock signal of interface modules. These trigger pulses are asynchronous with the actual clock of the module. This makes the testing of various combinations difficult. So it is necessary to synchronize the trigger pulse with a common clock. The proposed method uses a 1 MHz global clock to synchronize these trigger pulses and make them synchronous with the clock.

The synchronous design consists of 3 D-flip flops in master slave configuration. It is implemented for both rising edge and falling edge of the trigger pulse. The input given to the 1st flip flop is the trigger pulse. For the synchronization of falling edge of the trigger pulse the output of the 1st flip flop is inverted and AND-ed with the output of the 3rd flip-flop. The output obtained is the trigger pulse synchronized with 1 MHz clock. This synchronized output is given as the clock to the interface module. Fig. 2 shows the diagram of the proposed synchronous design technique.

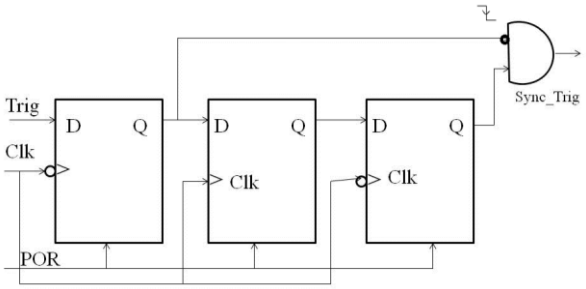

Fig. 2 Synchronous design technique to detect falling edge of the trigger pulse

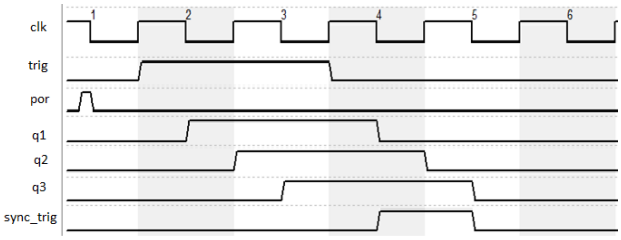

Fig. 3 Timing diagram of the synchronous design technique to detect the falling edge of the trigger pulse

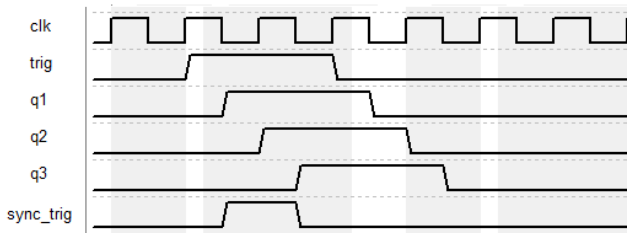

Fig. 4 Timing diagram of the synchronous design technique to detect the rising edge of the trigger pulse

To synchronize the rising edge of the trigger pulse same technique is implemented without inverting the output of 1st D-flip-flop and ANDing the inverted output of the 3rd D-flip-flop. Fig. 3 and Fig. 4 shows the timing diagram of synchronous design technique used to detect falling edge and rising edge of the trigger pulse respectively.

IV. DESIGN INTERFACES OF THRUSTER DRIVER MODULE

The design interface include the synchronous design of IO Decoder, Command generation, Serial digital channel interface, Thruster Driver interface, Thruster history monitoring Clock and Mode generation, Data ready interface and input-output ports [6].

A. IO Decoder

The IO Decoder generates active low IO Chip select signals CSn (24:0) and Rden1 signal for input ports using IO_DIS & M_ION and address lines IA (11:0). The input and output addresses are same and differentiated by Rdn and Wrn signal only [1]. The decoding address range is from 0100H to 0118H. The thrusters are operated between these address ranges.

B. IO Command Generation

The Command Decoder generates Internal Pulse commands with pulse widths 16ms, 64ms, 128ms &256ms

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

required for the firing of thrusters. These are decided by the Data-Bits D [10:11] and depending upon the data received from the Link1 or Link2 port [1]. The Link 1 and Link2 ports are the two ports present in the Bus Management Unit (BMU). They act as substitutes to each other. An 8-bit counter is used to generate the command pulses. Fig. 5 shows the command generation block.

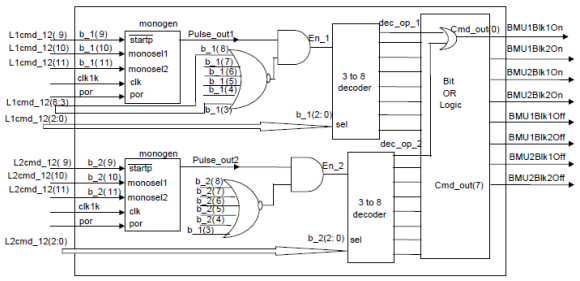

Fig. 5 Command Generation Block

The Pulse Command generator module gets triggered by the falling edge of Data-Bit b_1 (9) and b_2 (9) of link1 and link2 and generates Pulse command. This triggered pulse (Data-Bit) is synchronized with 1 MHz clock. A 3 to 8 decoder is used to decode the values of pulse command generated. The Link1 & Link2 module outputs are then ORed to generate the Pulse command outputs.

C. Serial Digital Channel Interface

The Serial Digital channel interface is to convert the Serial Digital data given to the thrusters into parallel data and route it to the input port [3]. The block is enabled when the Rdinh signal is high. The Serial Digital Data is shifted with the Selected Clock (Sel_Clk) of 40 KHz and Strobed to the output with the Selected Mode (Sel_Mode) for every 16 pulse. A 40 KHz clock is derived from the 1 MHz global clock.

D. Thruster Driver Interface

The outputs from the thruster driver interface are used to drive 16 thruster drivers [1] [4]. Fig. 6 shows the block diagram of Thruster driver interface.

It consists of 3 modes as follows:

- Direct Mode
- Timer Mode
- Serial mode

In direct mode, a 16-bit data is directly latched to drive the respective thruster drivers. The 16-bit data gives the information about which thruster drivers have to be fired.

In timer mode, depending upon 16-bit data, respective thrusters are fired for a particular duration of time. Timer consists of a 16-bit counter to fire the thruster for the required duration. It uses 1 KHz clock generated from 1 MHz global clock.

In Serial mode, wrn signal is synchronized with 40 KHz clock generated from 1MHz clock using synchronous design technique. Based on this a load is generated and the parallel data is converted serially using parallel to serial converter with master slave configuration.

Based on SelThr and Thr_Timer_en the different modes are selected to fire the required thruster drives.

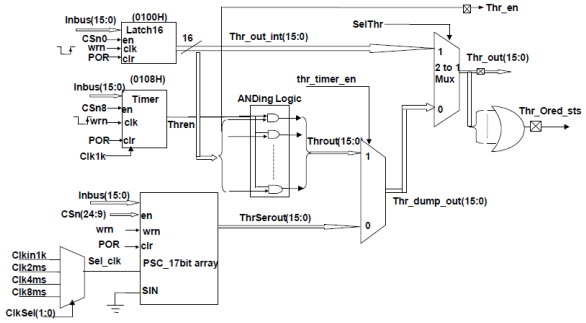

Fig. 6 Thruster Driver Interface

E. Thruster History Monitoring

The Thruster History monitoring interface consists of 16 8-bit counters [1]. It reads the status of the all the 16 thrusters and allows the counter to increment with selected clock frequency. The counter output is routed to processor data bus. Since selected clock is synchronized with the major cycle clock, as soon as major cycle flag is read, software should give his_mon_lp with in 1ms duration. Fig. 7 shows the block diagram of thruster history monitoring.

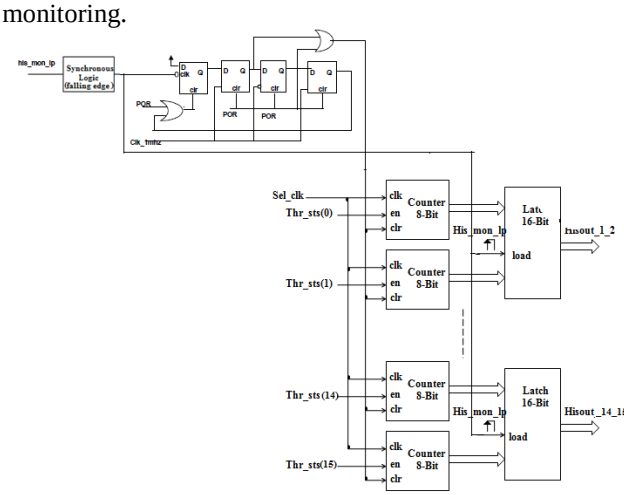

Fig. 7 Thruster History Monitoring

F. Data Ready Interface

The Data Ready type signals are edge sensitive. It is mainly used to detect if any signal coming to the module is a necessary pulse or a glitch. The qualifying pulse width of dtrdy qualifier is 187.5 us [1]. These are latched and routed to the input ports and cleared using output ports after being read by the processor.

G. Clock Generation

The Clock & Mode signals are required for receiving Serial Digital Data from Subsystems external to BMU [6]. This module generates the 1 KHz, 20 KHz, 40 KHz, 125 Hz, 250 Hz, and 500 Hz shift clock from 1MHz and Mode pulse every 16 clocks using 40 KHz clock. A counter is used in the clock and mode generation.

V. SIMULATION RESULTS AND DISCUSSION

Simulation is carried out using ModelSim V6.3f. The simulation results of thruster driver module using synchronous design technique shows the design is more reliable, efficient and can be best used for testability.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Fig. 8- Fig. 16 shows the Simulation results of design interfaces of Thruster Driver Module.

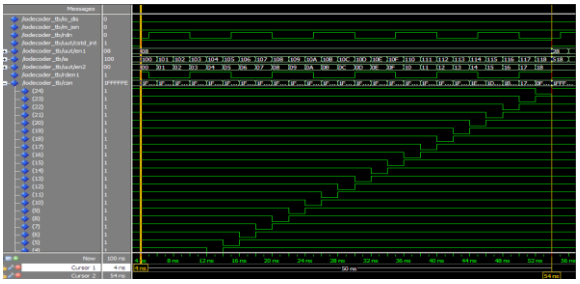

Figure 8: Simulation Result of IO Decoder

Fig. 8 shows the decoding of 25 chip-selects for the address range 100H to 118H which is required to operate the 16 thrusters.

Fig. 9 Simulation result of Command Generation

Fig. 9 shows the simulation result of command generation block which generates the command pulses of 16 ms, 64 ms, 128 ms and 256 ms duration for both Link1 and Link 2.

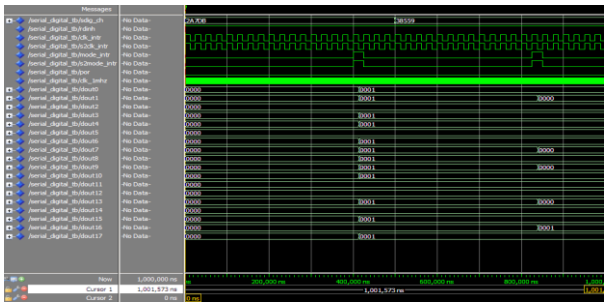

Fig. 10 Simulation results of Serial Digital Channel Interface

Fig. 10 shows the simulation output of SDCI. The input data given is 2A7DBh whose binary equivalent is 101010011111011011. The data given is latched to the output after the mode pulse is generated. The mode pulse is generated after every 16 clock pulses.

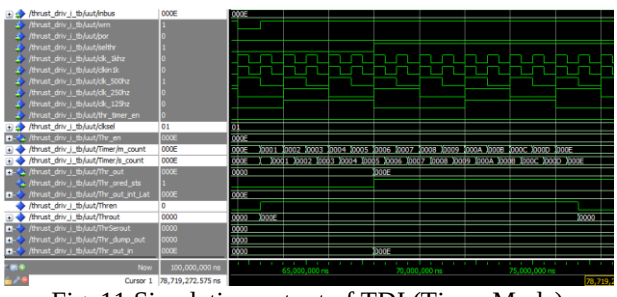

Fig. 11 Simulation output of TDI (Timer Mode)

Fig. 11 show the simulation output of TDI in timer mode. The input given is 000E. The thruster is fired "On" for the duration of 000E based on the clock selected. "Thren" signal is high for that particular duration which is latched to the output based on select signals.

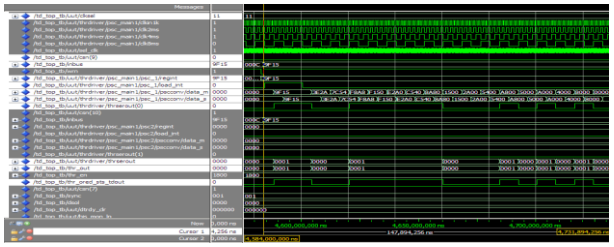

Fig. 12 Simulation output of TDI (Serial Mode)

Fig. 12 shows the simulation output of TDI in serial mode. The given input data is 9F15H. This given data is shifted out serially and the 16 thrusters are fired based on the data shifted. "Threserout" is the output where the 16-bit data is shifted out serially.

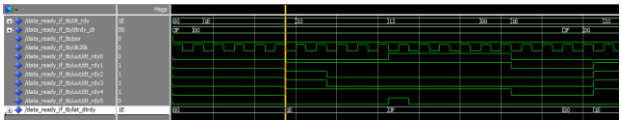

Fig. 13 Simulation result of data ready interface

In Fig. 13, the data ready input 1EH is latched to the output after the qualifier checks for the input signal of 3½ clock pulses.

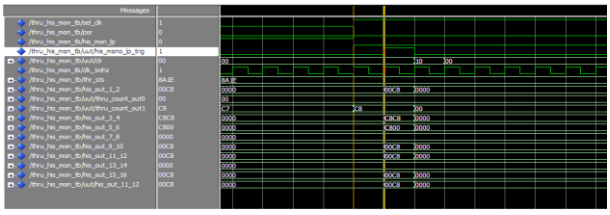

Fig. 14 Simulation result of thruster history monitoring

In Fig. 14 the falling edge of the history monitoring pulse is synchronized with the 1 MHz clock and the thruster status of 16 thrusters is latched to the output for a duration up to which the counter is counted.

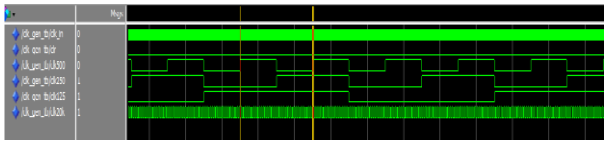

Fig. 15 Simulation results of the different clocks generated

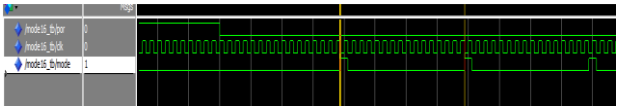

Fig. 16 Simulation result of mode generated for every 16 clock pulse

VI. SYNTHESIS RESULTS

Synthesis is carried out using Actel Libero IDE V 9.1. Parameters used to run synthesis are as follows:

Family : 54SXA
Device : RT54SX72S
Package : 624 CCGA/ CGA

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

TABLE II: COMPARISON OF SYNTHESIS RESULTS OF SYNCHRONOUS AND ASYNCHRONOUS DESIGN TECHNIQUE OF TD MODULE

	No. Of Core cells	**Area In %**	**Sequential cells**	**Combinational cells**
Asynch-ronous	3775 out of 6036	63	2258	1517
Synch-ronous	2482 out of 6144	41.12	1259	1223

Table II gives the comparison of synthesis results of synchronous and asynchronous design technique of TD module. The synthesis results of design of TD Module using synchronous design technique shows the number of core cells is being reduced from 63 % to 41.12 %, thus minimizing the area when compared to the existing asynchronous design technique.

VII. CONCLUSION

Study of thruster driver module is carried out in detail. The simulation results shows, the design is more reliable, efficient and can be best used for testability and the synthesis results shows the number of core cells are being reduced, thus minimizing the area when compared to the existing asynchronous design technique.

ACKNOWLEDGEMENT

The author would like to thank managements of ISRO Satellite Centre, Bengaluru and Jain University, Bengaluru for their fruitful guidance.

REFERENCES

[1] AIHS ASIC (ASIC#3) DESIGN DOCUMENT-ISRO-ISAC-TR-0839.
[2] Scott Hauck, Asynchronous Design Methodologies: An Overview. Proceedings of the IEEE, Vol. 83, No. 1, pp. 69-93, January, 1995.
[3] PI ASIC (ASIC#1) DESIGN DOCUMENT- ISRO-ISAC-TR-0837
[4] V. Lekshmi and Navle Sonal, "Hardware Realization of ASIC Based On-Board Computer", Journal of Spacecraft Technology, Vol. 22, No. 2, pp. 21–29, July 2012.
[5] Design document of AOCE for GEOSAT missions-ISRO-ISAC-GEOSATPR-1425
[6] B. K. S. V. L. Varaprasad and K.Parameswaran, "ASIC Development in ISAC", Journal of Spacecraft Technology, Vol. 21, No. 1, pp. 81-86, January 2011.
[7] Synopsys FPGA Synthesis Synplify Pro for Microsemi Edition-Reference, May, 2013.