Volume 2, No. 1, January 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

DESIGNING OF FOUR PORT CONTROLLED SWITCH-USING VERILOG

Chetan Sharma
(M.Tech-VLSI Department, JSS Academy of Technical Education, Noida, UP, India)
Chetan2042@gmail.com

Abstract: This paper gives a new type of switch design using veriology. Switch is an important part of VLSI chips. To making any RTL (Register Transfer Level) switch is required for transmitting the data through different ports. In this switch, control over transfer of data is also provided by using different set of registers as described in this paper. Another importance of this switch is the expandable properties by using reserve space for future use.

Keywords: Icarus tool, GTK wave, Gvim editor, RTL, Comparator cum receiver, Test bench

INTRODUCTION

Register Transfer Level (RTL) performs the sequence of register transfer in one clock or more than one clock cycle. It has two parts Data subsystem and Control subsystem. For any data transmission two things requires firstly set of registers secondly control of sequencing of that registers as shown in fig (a).

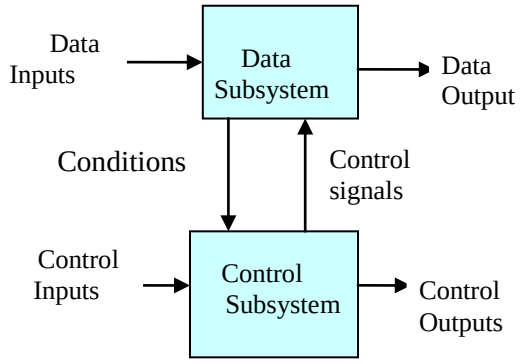

Fig (a): Structure of RTL system

For design any VLSI chip firstly RTL of that chip's specification is designed then test bench is written using HDL (Hardware descriptive language) either VHDL or Verilog for testing the functionality of the chip. After this frontend part the whole code is given to Backend team for fabrication process.

Four port switch is an important component of any design for transmitting data through selecting a particular port depending on the bits of incoming data. It gives all the signals such as Ready, Read, Detain, Dataout for reliable transmission of data. All the designing is done here by Icarus tool using verilog with GVIM editor. Wave form is gotten by the tool GTK wave.

REGISTER SETS

There are four register sets in the architecture of this switch:

DA: It is 8 bit register which use to store the destination address of data. Last two bits of DA decides that which port will be use for transmission of 8 bit data as shown in fig (c) and all other 6 bit is reserve for future use. These blank spaces can be programmed according to user requirement.

SA: It is 8 bit register which is used to store the source address of Input data.

LENGTH: It is 8 bit register which is used to decide that how many number of byte will have to be transmitted. After completing the number of byte read from Length register, transmission is stopped.

CONTROL: It is also 8 bit register which is used to control the flow of transmission. If 0^{th} bit is 0 then no operation will be performed else transmission will happen.

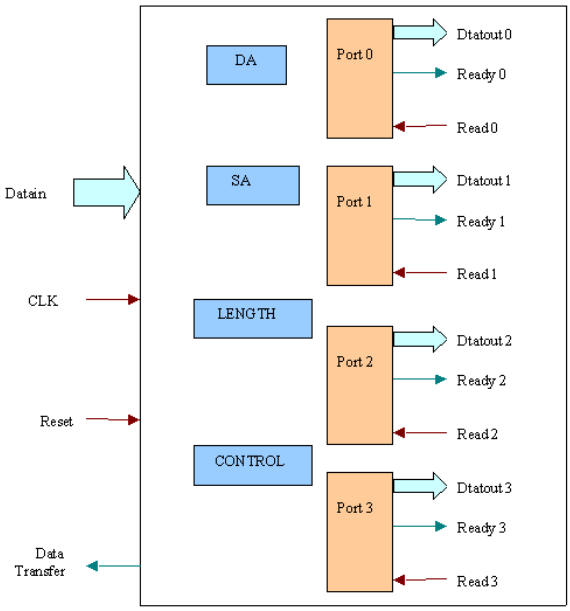

Fig (b): Block diagram of four port switch

Bit 1	Bit 0	Selected PORT
0	0	Port 0
0	1	Port 1
1	0	Port 2
1	1	Port 3

Fig(c): Port selection by DA register

SIGNALS DESCRIPTION

1. **Datain**: This is 8 bit input data to switch.
2. **Dataout 0**: This is 8 bit output data from port 0.
3. **Dataout 1**: This is 8 bit output data from port 1.
4. **Dataout 2**: This is 8 bit output data from port 2.
5. **Dataout 3**: This is 8 bit output data from port 3.
6. **CLK**: This is 1 bit clock signal to switch.
7. **Reset**: This is 8 bit register which is used to reset all the register sets of the switch.
8. **Read 0**: If this signal is 1 only then ready 0 can high for transmission of data.
9. **Read 1**: If this signal is 1 only then ready 1 can high for transmission of data.
10. **Read 2**: If this signal is 1 only then ready 2 can high for transmission of data.
11. **Read 3**: If this signal is 1 only then ready 3 can high for transmission of data.
12. **Data transfer**: This is 1 bit signal which is used to giving the conformation of the completely transmission of data.

WORKING

When the 8 bit data comes to switch by Datain signal then this byte is stored in DA register which decides the port by which it has to be out. Second byte comes to SA register which tell the source address of the input data. Third byte of input goes to LENGTH register which decides the number of byte to be out via port. Fourth byte comes to CONTROL register which decides whether the data should transfer or not. Now if read signal of corresponding port is 1 then ready signal of same port become high. Then next coming bytes will transfer to output 0 or output 1 or output 2 or output 3 depending on first four bytes. On completing the transmission data transfer signal become high.

DESIGNING STEPS

The Designing is done in following steps:

(a) RTL Design:
In the RTL Design all the functionality of switch is defined using verilog as described in working. At every posedge of clock or reset datain comes to each register and port by using counter. Counter is increased by 1 after every register.

(b) CLK Reset generator:
This module generates the system clock and reset pulse for whole system i.e. RTL, Packet generator, Comparator.

(c) Packet generator:
It generates all the input signal for RTL and output signal for comparing with output of RTL in the receiver.

(d) Comparator cum receiver:
Finally decides whether RTL is working properly or not by comparing output of RTL and Packet generator.

(e) Top switch:
It generates the dump file and another function of top switch is to connect all the signals of each blocks.

(f) Test case:
By the test case input value is given for testing the design.
Now a single list is form of all above module for compiling.
The total design flow of controlled Switch to testing blocks is given by following fig (d). All the signals are connected according to programming concepts. After a.out file wave is seen by GTK wave tool.

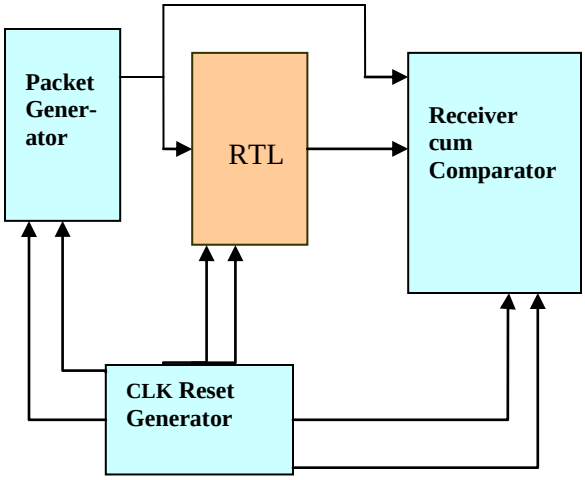

Fig (d): Testing of RTL switch

SIMULATION RESULTS

This is the simulation result of RTL of the switch.

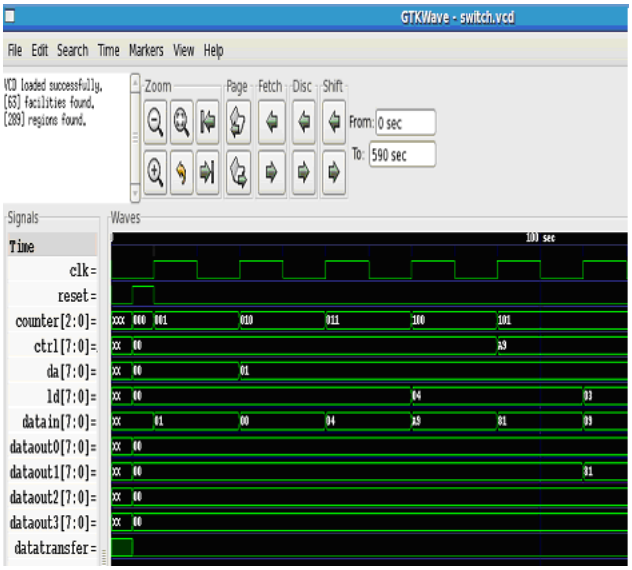

Fig (e): Simulation result of RTL

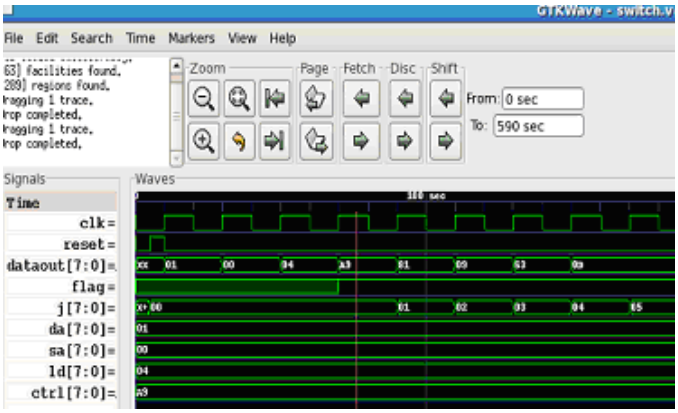

Fig (f): Packet generator waveform

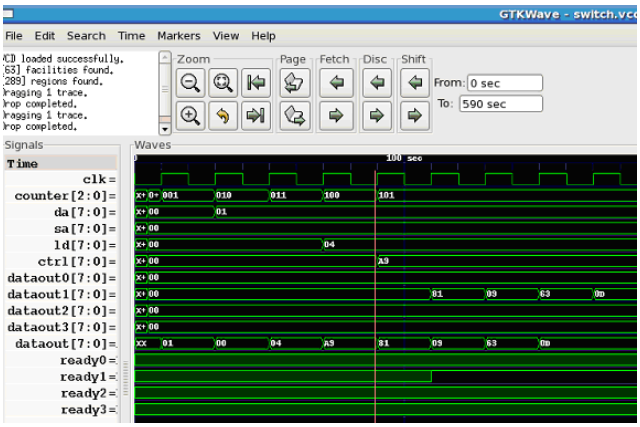

Fig (g): Comparator cum receiver wave form

CONCLUSION

In this paper the data transfer through controlled switch is designed using Verilog. It has minimum delay and high controllability on transmission of data from source address to destination address because of using Length and Control register. This design also provides the less silicon area consumption due to the reason of simple architecture.

REFERENCES

[1] Flynn, D., "AMBA: Enabling Reuseable On-Chip Designs", IEEE Micro, 17(4**)**, July/August 1997, pp 20-27.

[2] J. Liang, Swaminathan, S, "ASOC: a Scalable, Singlechip Communications Architecture", Tessier, R. Parallel Architectures and Compilation Techniques, 2000, Proceedings. International Conference,2000, pp 37-46.

[3] P. Laramie, "Instruction Level Power Analysis and Low Power Design Methodology of a Microprocessor", MS Thesis, University of California at Berkeley, 1998

[4] AMBA Specification (rev2.0) and Multi Layer AHB Specification, Arm: http://www.arm.com, 2001

[5] J. Becker, L. Kabulepa, F.M. Renner, M. Glesner,"Simulation and Rapid Prototyping of Flexible Systemson-a-Chip for Future Mobile Communication Applications", RSP 2000, pp. 160-165.

[6] Reconfiguring Excalibur Devices under Processor Control: Application note 298, Feb 2003, ver 1.0.http://www.altera.com

[7] http://www.xilinx.com

[8] http://www.altera.com/products /ip/processors/nios/niondex.html

[9] C8237 Programmable DMA Controller Core, http://www.cast-inc.com.

AUTHOR

Chetan Sharma received the B.Tech (Electronics & Communication Engineering) degree from Uttar Pradesh Technical University, Lucknow in 2009, pursuing M.Tech (Advance Electronics & Communication specialization in VLSI Design) from JSS Academy of Technical Education, Noida, India 2009-11 batch.