A 16 Bit RISC Microprocessor Designing

Amit
M.Tech Student(EC + VLSI)[1]
Suresh Gyan Vihar University

Vipin Kumar Gupta
Assistant Professor[2]
Suresh Gyan Vihar University

Abstract:- We are working on a paper which name 16 bit microprocessor designing By VHDL with better performance. This paper is based on a new low power designed technique that allows us to solving most of the problems. By use of only pipelining. We can implement more complex circuits of digital electronics .this method are beneficial for fast low power circuits by use of reduce number of transistors. We are reducing the delay of our new microprocessor by use VHDL language. In this thesis we will work at delay and power consumption part. We are reusing the delay up 6.00ns . Basically delay will depends on a total number of Lut's . If we will reduce the total number of LUT the n it will automatically reduce the delay. After reduce the delay we are increasing the speed of a controller. Actually if speed increase will found in IC then the performance of any circuit in which that IC is using also increase.

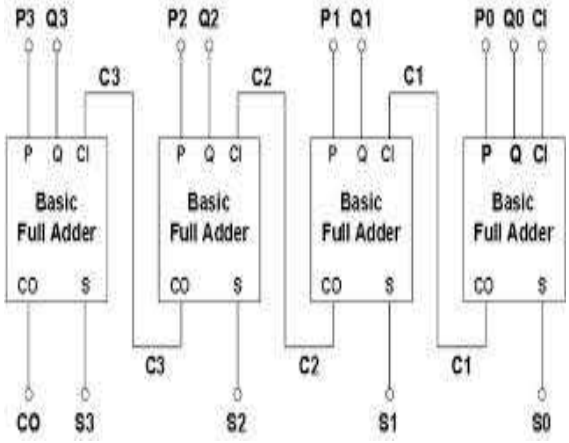


Fig 1 Serially full adder connection

INTRODUCTION

We are using pipelining for reducing the power consumption and area with number of LUT'S .Pipelining is a method in which we will transfer all the data in parallel way so that no one data will wait for the competition of any other it will goes parallel

During this if we will use a low delay and power consumption controller then it will increase the performance of the circuit in which it is using.
we are increasing the speed of a controller. Actually if speed increase will found in IC then the performance of any circuit in which that IC is using also increase. During o this if we will use a low delay and power consumption controller then it will increase the performance of the circuit in which it is using. In this technique we will transfer data to destination in parallel way. Regarding this all the data will go parallel no one will wait for the completion of any other process.

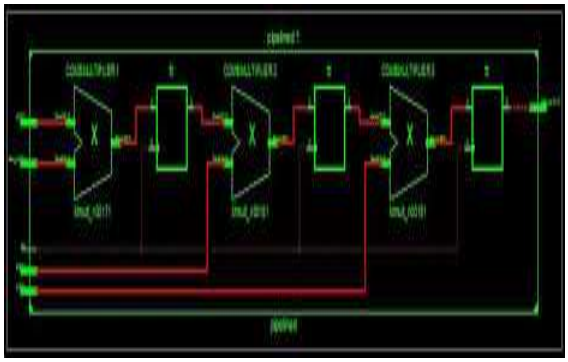


Fig 2 parallel connection of full adder

The number of instruction is also increasing. We have 4 bits for instructions. We can use 16 instructions.

Table 1

For instructions have four bits	For data memory address 4bit
7 3	0

generic (nop:std_logic_vector(3downto 0):="0000";

mult : std_logic_vector(3 downto 0) := "0001";

add:std_logic_vector(3 downto 0):="0101"
sub:std_logic_vector(3 downto 0):="0110";

input:std_logic_vector(3 downto0):="0111";

```
oput:std_logic_vector(3 downto 0):="1000";
lda:std_logic_vector(3 downto 0):="1101";
ldb:std_logic_vector(3 downto 0):="1111";
```

We declare all the mnemonics in the entity . We are using generic statement which is using for globally constant declaration . I am saying that all the mnemonics will be declared for globally program. We have been given a particular address to men monies so that w can call to them according to their address.

There are total eight mnemonics which is using in microprocessor .

NOP :- no operation

MULT :- For multiplication

ADD :- For addition

SUB :- subtraction

Input :- for transfer input

Output :- take a output

LDA :- load in accumulator A

LDB :- Load in accumulator B

```
process(clear)
        begin
        if(clear='0')then
        line1<=lda & four;
        line2<= mult & three;
        line3<= mult & three;
        line4 <= oput & acca ;
        else
        line1<="ZZZZZZZZZZZZZZZZ";
        line2<="ZZZZZZZZZZZZZZZZ";
        line3<="ZZZZZZZZZZZZZZZZ";
        line4<="ZZZZZZZZZZZZZZZZ";
        line5<="ZZZZZZZZZZZZZZZZ";
        line6<="ZZZZZZZZZZZZZZZZ";
        line7<="ZZZZZZZZZZZZZZZZ";
        line8<="ZZZZZZZZZZZZZZZZ";
 end if;
        end process;
```

This is a part for user coding. In this part user can perform the coding according to their requirement. Microprocessor gives flexibility to user he / she can arrange a coding according to our behalf. Like we have written that in line 1 that LDA & four. This means is we want to save data four in accumulator A Now in second line we have declare that MULT & three . This means user want to multiply three to four. Now again mult three that means a result which comes after multiply by three or four is again multiply by three. OPUT Acc a that means we want to take a output at a output port.

RESULTS

We are using pipeline concept to remove this problem . we are also find a our another problem that this controller will work slow because it have more delay according to my point of view . We can design more better microprocessor by VHDL. We have designed a new controller by pipelining which have better output from this paper . . We are showing results in waveform , power consumption , delay .

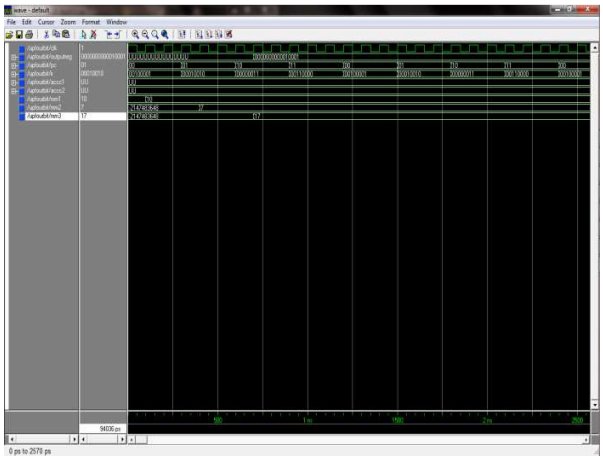
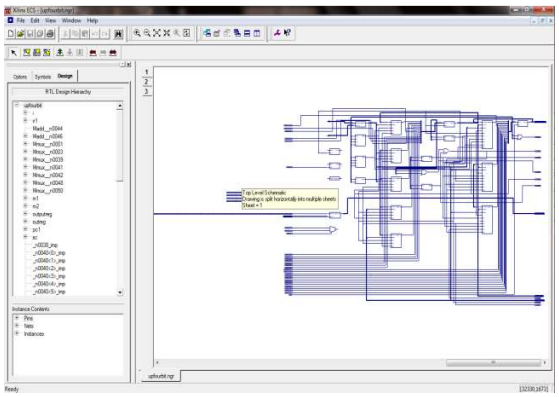


Fig 3 Waveforms for the output

We are giving our results in waveform. If we will view of this RTL it will be shows that there are many few LUT’s which is showing working of controller. this is a main reason so that we are going at low delay and power consumption .In this waveform at a output port a result is coming which is final

result of microprocessor . Remaining all are the signals which is using for interconnection of pipelining. We have been given a flexibility to user that he can arrange a coding according to our choice. We have been given a part for coding where user can perform our coding .



RTL1 view

This is a RTL view one part which is showing that there are many few LUT's are using . Now we can show that our delay have been reduced . We have been reduced our delay nearby 30 %.

Power consumption for FPGA sparten 3e XC3S50 is coming 0.038 W. For this we are using FPGA sparten series because at a paper we are working that is based on sparten 3e series so we are taking all the parameters at sparten 3e series of FPGA.

Table 2

Element	No of slices	Total no of slices	%
No of slices	15	192	7%
No of slices flip flop	20	384	5%
No of 4 input LUTs	24	384	6%
No of bonded IOBs	17	86	19%
No of GCLKs	1	4	25%
No of IOs	17		

CONCLUSION ANS FUTURE SCOPE

We have design a new microprocessor in which we have designed own mnemonics. We have a authority to give any name of mnemonics because we are designing a self depend new microprocessor. This controller user has a flexibility that he/she can perform any coding in this controller. For example we want to perform 3 * 3 * 3 then we can write a program in assembly language controller gives a output. User can change the program if user want to perform 3-2+1 then user can perform this. For perform user want to write a program in assembly language and then program will be performed.

In this thesis we worked on parameters of controller. We have a lot of problem in delay. Suppose we are making a program for led blinking but when we implement it at hardware due to this there will be a some delay which will be coming in hardware. That delay is non removable. Because this is a delay of IC . We never remove the delay of IC. Microprocessor is

Using mostly in every circuit of electronics. So if we reduce the parameters of controller then that will be shows that we are increasing speed for that entire component in which it is using.

In this thesis we are working at delay, power consumption and area for microprocessor designing. our results are coming better from previous work . The delay is coming 7.380ns (3.480 logic, 3.90 ns route) (47.2% logic, 52.8% route) for 16 bit microprocessor. Here we are finding total LUT's 38 which is less from previous work.

REFERENCES

[1] M. A. S.D. Poz, J. E. A. Cobo, W. A. M. V. Noije and M. K. Zuffo, "A Simple RISC Microprocessor Core Designed for Digital Set-Top-Box Applications," IEEE International Conference on Application-Specific Systems, Architectures, and Processors, 2000. Proceedings, pp. 35-44.

[2] Xia Li, Longwei Ji, Bo Shen, Wenhong Li and Qianling Zhang, "VLSI Implementation of a High-performance 32-bit RISC Microprocessor," International Conference on Communications, Circuits and Systems and West Sino Expositions, IEEE 2002, Vol. 2, pp.1458-1461

[3] Sivarama P. Dandamudi, "Fundamentals of Computer Organization and design" Springer-Verlag New York, Inc., 2003.

[4] John L. Hennessy and David A. Patterson, "Computer Architecture A Quantitative Approach," 3rd Edition, Morgan Kaufmann Publishers, 2003.

[5] Zhenyu Gu, Zhiyi Yu, Bo Shen and Qianling Zhang, "Functional Verification Methodology of a 32-bit RISC Processor," International ference on Communications, Circuits and Systems and West Sino Expositions, IEEE 2002, Vol. 2, pp. 1454-1457.

{6] Peter J. Ashenden, "The Designer's Guide to VHDL," 2nd Edition,
Morgan Kaufmann Publishers, 2002.
[7] Spartan-Il Platform FPGA handbook October 24, 2002.