

# Enhancing Frequency Of Operation of A Single Cycle CPU Using Few Synthesis Options In Xilinx

Swati G Tenglikar<sup>1</sup>, Anand C<sup>2</sup>

<sup>1,2</sup>Dept of Electronics and Communication

<sup>1,2</sup> KLE Technological University Hubballi, Karnataka, India

**Abstract-** A clock cycle is defined as the time from one rising edge to the next rising edge. A synchronous computer is one in which all operations are controlled by a clock. A single-cycle CPU (central processing unit) executes each instruction in one clock cycle. CPU states (program counter (PC) and registers) are updated at the rising edges of the clock. Compared to other CPUs, the single-cycle CPU is the most cost-effective but time-consuming CPU. The Single Cycle CPU consists of an ALU, Register File, Control Unit, Data Memory and Instruction Memory. The instruction format of the instructions is also explained. The whole design is split into smaller modules. Each module is designed using Verilog HDL. The synthesis and simulation results are obtained using Xilinx 14.2v (Device: Artix7).

**Keywords-** CPU, clock cycle, MIPS (Millions Of Instructions Per Second), PC, Synchronous computer

## I. INTRODUCTION

The Single Cycle CPU (SCCPU) executes each instruction in one clock cycle. The machine must operate at the speed of the slowest instruction. The instruction cycle consists of the steps Instruction Fetch, Instruction Decode and Instruction Execution. These are performed by the hardware circuits. The instruction must be fetched by the processor from the instruction memory. There is a Program Counter (PC) in the CPU that points to a memory location. The instruction at that location is executed by the CPU. If the fetched instruction is neither a branch instruction nor a jump instruction, the PC will be incremented by 4, pointing to the next instruction. A multiplexer is used to select the next PC, which will be then written to the PC at the next rising edge of the clock. The instruction is decoded with the help of the Table 1 and the corresponding values of the fields in the instruction are considered for execution. The main components for executing an instruction are Arithmetic and Logic Unit (ALU), Register File and a Control Unit. The paper explores how the max operating frequency could be improved for the same design using the Synthesis options in Xilinx.

## II. IMPLEMENTATION

The schematic of SCCPU is shown in the Figure 1 below. The various components of the SCCPU are the ALU, Control Unit, Register File, Data Memory and the Instruction Memory. A 4:1 MUX is used to select the PC contents depending on the type of the instruction. Five 2:1 MUX are used to select the ALU inputs, Memory output, data inputs of the Register File, and the Write register number of the Register File. The instructions considered in the project are the MIPS32 instructions which are 32 bits in length. The various instructions that are implemented in the project are shown in the Table 1.

Table 1: MIPS Integer Instruction<sup>[1]</sup>

Inst	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	Meaning
add	000000	rs	rt	rd	00000	100000	Register Add
sub	000000	rs	rt	rd	00000	100010	Register Subtract
and	000000	rs	rt	rd	00000	100100	Register AND
or	000000	rs	rt	rd	00000	100101	Register OR
sll	000000	0	rt	rd	sa	000000	Shift left
addi	100000	rs	rt	imm			Immediate add
lw	100011	rs	rt	offset			Load memory word
sw	101011	rs	rt	offset			store memory word
lui	001111	00000	rt	imm			Load upper immediate
jal	000011		addr				Call

The ALU module consists of two 32-bit inputs, a 32-bit output to indicate the result, a 1-bit output which acts as a zero flag to indicate if the result is zero and a 4-bit control signal which selects among the various operations that could be performed using the ALU.

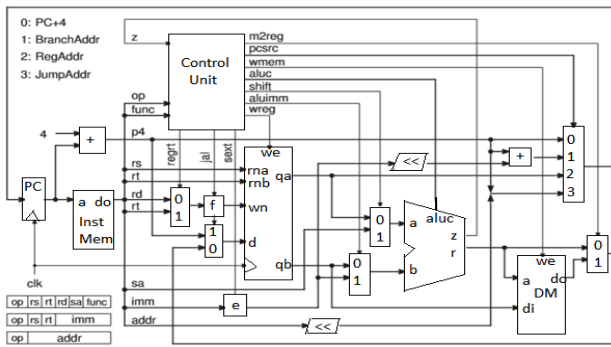


Figure 1: Schematic of Single Cycle CPU

There are 32 general purpose registers in the Register File. The register file has two read ports and a write port. The ports are 5-bit each. The register file also has a 32-bit input data and a write enable which is used to write to the register. The Control Unit is responsible for the functioning of the processor. The various operations that should be performed are controlled by the control unit. The Control Unit generates the signals based on the current instruction that is being executed. The instruction memory is implemented as a 32 bit ROM. The ROM is basically a combinational design. The combinational blocks are designed to take care that no latches are inferred.

### III. RESULTS

The Verilog code was synthesized using the Xilinx 14.2v software. The device used for the synthesis is Artix 7. The simulation result of the design was obtained through a testbench. The testing was done by splitting the design into modules and writing testbench for each module separately. The instructions are written into the instruction memory, fetched by the CPU from the instruction memory, and then executed. The Figure 2 below shows the Timing Summary of the worst path in the design.

```

Timing constraint: Default period analysis for Clock 'clk'
Clock period: 6.984ns (frequency: 143.193MHz)
Total number of paths / destination ports: 2001304 / 385
-----
Delay: 6.984ns (Levels of Logic = 11)
Source: A/pc_2_1 (FF)
Destination: A/pc_31 (FF)
Source Clock: clk rising
Destination Clock: clk rising
-----
Data Path: A/pc_2_1 to A/pc_31
-----
Cell:ln->out fanout Delay Delay Logical Name (Net Name)
FDR:c->q 1 0.398 0.780 A/pc_2_1 (A/pc_2_1)
LUT5:10->0 60 0.105 0.734 IM/Mmux_inst61 (Inst_18_0BUF)
LUT5:12->0 3 0.105 0.483 A/A/B/C/L/Mmux_k_451 (A/A/B/C/L/Mmux_k_45)
LUT6:14->0 3 0.105 0.369 A/A/B/Sh161 (A/A/B/Sh16)
LUT6:15->0 2 0.105 0.362 A/A/B/Sh5611 (A/A/B/Sh561)
LUT6:15->0 1 0.105 0.357 A/A/B/J/K9/Mmux_temp344 (A/A/B/J/K9/Mmux_temp343)
LUT6:15->0 1 0.105 0.357 A/A/B/J/K9/Mmux_temp346 (A/A/B/J/K9/Mmux_temp345)
LUT6:15->0 3 0.105 0.792 A/A/B/J/K9/Mmux_temp347 (r_24_0BUF)
LUT6:11->0 1 0.105 0.357 A/A/B/J/z5 (A/A/B/J/z4)
LUT6:15->0 1 0.105 0.357 A/A/B/J/z8_SW1 (N20)
LUT6:15->0 30 0.105 0.486 A/A/A/pcsrc<0>1 (A/pcsrc<0>)
LUT5:14->0 1 0.105 0.000 A/pcout<28>1 (A/pcout<28>)
FDR:D 0.015 0.015 A/pc_28
-----
Total 6.984ns (1.568ns Logic, 5.416ns route)
(22.5% Logic, 77.5% route)
-----
    
```

Figure 2: Timing Summary

The maximum frequency obtained through this design is 143.19MHz [i.e, clock period of 6.984ns]. This frequency is obtained with the default Synthesis options.

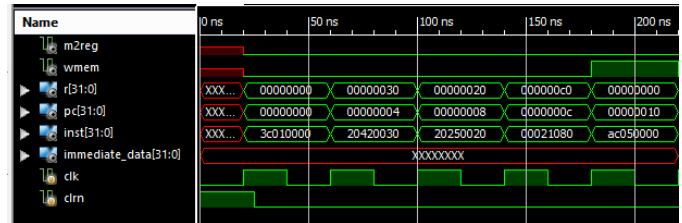


Figure 3: Simulation Waveform

The simulation waveforms show a 32 bit instruction given to the design. The design performs the required operation and generates the PC value and the 32 bit result. The design is first reset by the high on *clrn* signal. A memory write operation is indicated by a high on *wmem*. In case the operand is an immediate data, it is fetched from the data memory. For instance, the instruction code 00021080 (*sll \$2, \$2, 02*)<sup>[1]</sup> in Figure 3 shifts the data to the left as indicated in the waveform. The Figure 4 depicts instructions when an immediate data is given as an operand. The unknown values depicted in the Figure 3 and Figure 4 by red blocks just indicates that the corresponding instruction is performed without immediate data.

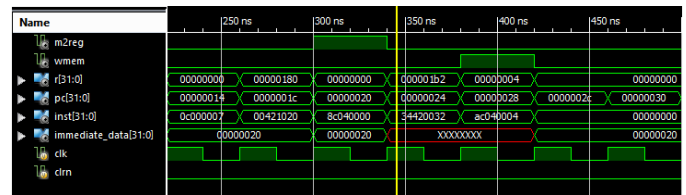


Figure 4: Simulation Waveform cndt.

### IV. IMPROVEMENTS

The Xilinx 14.2v tool is further explored to improve the design operating frequency. The various Design Goals and the Synthesis options give a wide range of constraints such as high speed, low power, less areawhich could be applied to the design to obtain an optimal clock period. However, if the design is subjected to various constraints at the same time then the design QOR is not always guaranteed to be optimal. Thus the constraints must be thoughtfully applied and the tool must be allowed to choose few constraints by itself for the optimal results. A Synthesis of the design with a fanout of 30 vs the default 100000 [Table 2] didn't show any appreciable difference in the max frequency. A fanout of 30 was chosen to be a more conservative estimate.

Table 2: Improvement based on fanout &amp; IOB constraints

Trial	Synthesis Options	Max Frequency (MHz)
1	Default Synthesis	143.19
2	Synthesis with : Optimization Goal = Speed	151.39
3	Synthesis with : Optimization Goal = Speed AND Fanout = 30	150.17
4	Synthesis with : Optimization Goal = Speed AND IOB-Packing 'ON'	139.56
5	Synthesis with : Optimization Goal = Speed AND IOB-Packing 'ON' AND Fanout = 30	140.42

Another constraint is the IOB packing which allows a flip-flop or latch to be moved into IOBs when the device is mapped. Though the pad to setup time decreases, the delay from the IOB latch/flip-flop to the next synchronous element increases which is also evident from the Table 2. The default value of IOB packing is Off.

As seen - the Tool is able to optimize the speed of the design more with fewer given constraints [as in Case 2] than when more [i.e. IOB/Fanout] limitations are imposed [Cases 3, 4 & 5]. Here a maximum frequency of 151.39MHz could be achieved (i.e. clock period of 6.659ns) with just the Optimization Goal set to 'Speed'.

## V. CONCLUSION

A few Synthesis options were explored to improve the design QOR. The CPU thus designed operates at a maximum frequency of 151.39MHz (6.659ns) which is comparatively better than the maximum frequency of 143.19MHz (6.984ns). This improvement was due to the Synthesis constraints (high speed, IOB packing) applied to the design. Other synthesis options such as Power reduction, Synchronous/Asynchronous Set / Resets could be explored for further enhancements.

## REFERENCES

- [1] "Computer Principles and Design in Verilog HDL", Yamin Li, Hosei University, Japan, Published by John Wiley and Sons Singapore Pte Ltd, 1st Edition 2015
- [2] "Complete Digital Design", Mark Balch, McGraw-Hill Publications, 2003
- [3] "FPGA Prototyping by Verilog Examples", Xilinx Spartan-3 Version, Pong P Chu, Cleveland State University, A John Wiley and Sons, Inc., Publications, 2008
- [4] "Digital VLSI Design with Verilog", John Williams, Silicon Valley Technical Institute, 2008