Turkish Journal of Computer and Mathematics Education *Vol.11 No.01 (2020), 637-641*

Research Article

Design of Efficient High Speed Adders For 4-BIT Microprocessor

Mr. Monty Kumar
[1,2]Guru Kashi University, Talwandi Sabo

ABSTRACT

The building of a Verilog model of a 4-bit microprocessor from fundamental gates and the design of high-speed adder circuits using Xilinx ISE 8.1i are the main topics of this article. To examine the design parameters, the designs are executed on a Field Programmable Gate Array (FPGA). The reason for this inquiry is because an adder is a basic building element of the Arithmetic Logic Unit (ALU) and would be a limiting factor in the Central Processing Unit's performance. We used Verilog and Xilinx ISE 8.1i to simulate and synthesise several adders such as the Ripple Carry Adder, Carry Look Ahead Adder, and Carry Skip Adder in this research work. The architecture's Verilog codes were created with Xilinx ISE 8.1i and aimed for the Xilinx Spartan 3E FPGA. The functioning of high-speed adders is examined, as well as factors such as area and speed.

Keywords: Carry Look Ahead Adder, Carry Skip Adder Field Programmable Gate Array, Ripple Carry Adder.

I. INTRODUCTION

A microprocessor is a programmable, clock-driven, register-based electrical device that reads binary instructions from a memory device, receives binary data as input, processes data according to those instructions, and outputs the results. The model in this study is created in such a way that it can run an assembly-language programme written with the instruction set. We started by designing the processor using Ripple Carry Adder [1] in this article. Carry Look Ahead and Carry Skip Adders were created in the following stage of development, as well as new 4-bit processing units featuring these high-speed adders [6]. We created a 4-bit processor with ease of design in mind, and this unit can easily be upgraded to 8-bit, 16-bit, 32-bit, and so on.

II. DESCRIPTION

The design of a 4-bit microprocessor is completed in steps [2]. The many steps of the design process are outlined here.

2.1 GATE Implementations:

The behavioural descriptions of the XOR, AND, OR, and NOT gates are written and evaluated on test benches at this point.

2.2 The 4-bit Full Adder:

In this step, a top-bottom or bottom-up structural description of a 4-bit complete adder is written and confirmed in a test bench. The first-stage gates are used to create the adder. The adders accept two 4-bit numbers as inputs and outputs, and they take two 4-bit numbers as inputs and outputs. The fundamental problem with this adder is that carry must propagate through all stages, resulting in a longer propagation delay.

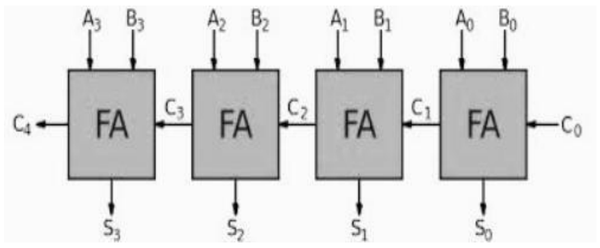

Figure 2.1: Ripple Carry Adder

2.3 The 4-bit ALU:

There are three selection bits: S2, S1, and S0, allowing for a total of eight operations. It employs a stage 2 adder. The Arithmetic and Logic Unit (ALU) is a processor's basic building component that conducts arithmetic and logic operations. Table 2.1 lists the features and functions. Three selection bits, two 4-bit integers, and cin for adder are inputs to the ALU, which outputs a 4-bit number, f, and cout [4]

S2	S1	S0	Cin	F	Description
0	0	0	0	f = a	Transfer a
0	0	0	1	f = a + 1	(b=0000)
0	0	1	0	f = a + b	Increment a
0	0	1	1	f = a + b + 1	Add b to a
0	1	0	0	f = a + (not b)	Adds with carry
0	1	0	1	f = a + (not b +1	Subtract with
0	1	1	0	)	borrow
0	1	1	1	f = a -1	Subtract
1	0	0	X	f = a	Decrement a
1	0	1	X	f = a or b	Transfer a
1	1	0	X	f = a xor b	(b=1111)
1	1	1	X	f = a and b	OR
				f = not a	XOR
					AND
					NOT

Table 2.1 ALU operations

2.4 The Registered ALU:

This paper introduces clocked register modelling methodologies. The ALU of stage 3's output signals f and cout are synced to the rising edge of the clock. The initialization of these output signals is controlled by an active low asynchronous reset signal.

2.5 The Instruction Decoder:

The instruction decoder is a CPU component that translates bits contained in the instruction register or in CPUs into control signals that control other CPU components [3]. This module is a state machine that accepts an instruction set that consists of a 3-bit opcode and two 4-bit operands as input. The instruction set is listed in table 2.1 below.

2.6 16 X 4 RAM Model :

A random-access memory device permits data elements to be read and written in about the same amount of time regardless of the order in which they are accessed.

The input ports are :

addr (4-bit vector), datain (4-bit vector)

csn (1 bit): chip select, it is 0 when memory is in use

rwn (1 bit) : 1 = read memory; 0 =write memory

The output port :dataout (4-bit vector) : Output data from RAM when in read mode.

2.7 The 4-bit Processor :

The RAM, ALU, and Instruction decoder modules are all instantiated once the dsp-processor module is constructed. Everything is arranged in the manner seen in Figure 2.1.

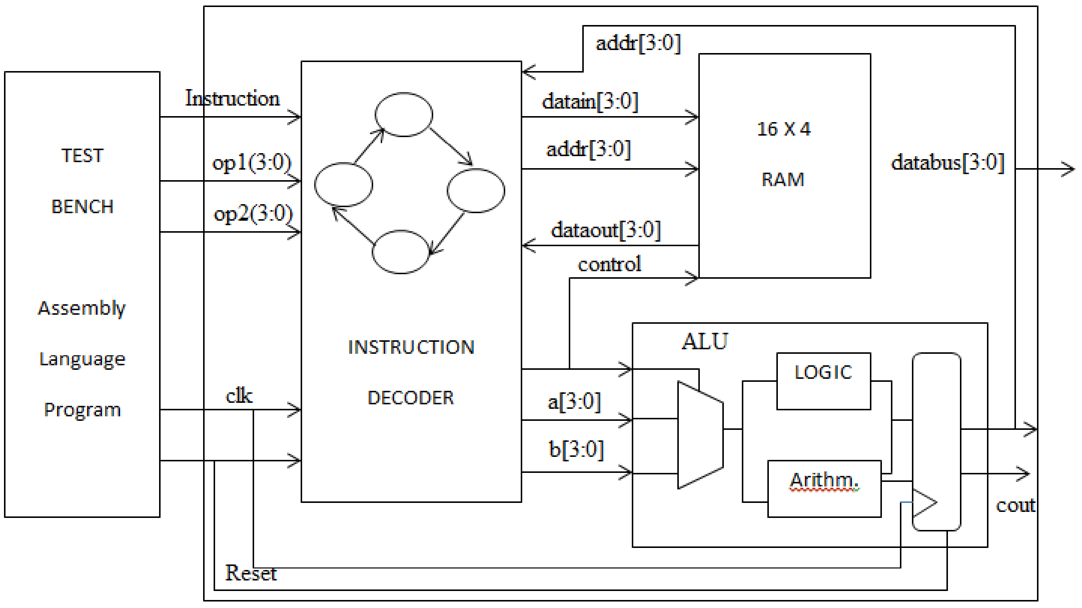

Figure 2.1: Block Diagram of 4-bit Microprocessor

2.8 Assembly Language Programs :

A file holding the processor's assembly language software is produced. A tiny C/C++ compiler is being constructed to convert this programme into a test-bench programme in which the instructions are fed into the instruction decoder in figure 2.1 with the appropriate durations between them.

2.9 Synchronous RAM with Bidirectional Data :

In the RAM model that has previously been created, a clock and reset signal will be included. To sync the data input and output with the clock, the code must be updated. In order to create an in out signal, the datain and dataout signals must be mixed. Four RTL-style tristate drivers will now be in charge of this bidirectional bus.

III. DESIGN OF ADDERS

So far, we've finished designing a 4-bit processor that includes an ALU with Ripple Carry Adder, RAM, and an Instruction Decoder. Now we'll work on making more efficient adders. Carry Look Ahead Adder and Skip Adder with you.

3.1 Carry Look Ahead Adder*:*

Because the time to add two integers relies on the logarithm of the size of the operands, the CLA adder is potentially one of the quickest systems for doing so. The ideas of generating (G) and propagating (P) carries are used in carry look ahead reasoning. For each bit location, it works using two signals termed P and G.
The P and G are shown below:
Ci+1 = Gi+ PiCi here Gi=AiBi and Pi= (Ai $\oplus$Bi)
Si=Ai $\oplus$ Bi $\oplus$Ci = Pi $\oplus$Ci.

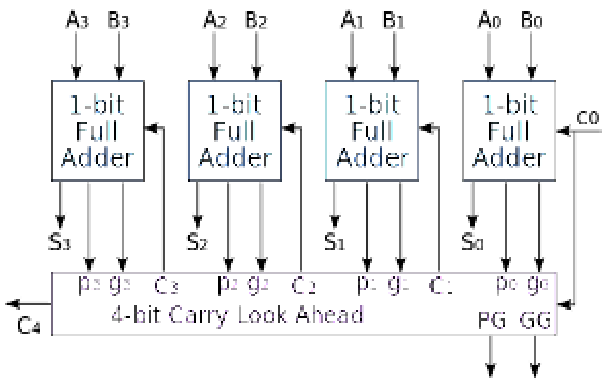

Figure 3.1: Carry Look Ahead Adder

The total and carry from the ith complete adder are represented by Si and Ci+1, respectively. The carry look ahead adder is divided into two modules: (1) the partial full adder, which creates Si, Pi, and Gi; and (2) the full adder, which generates Si, Pi, and Gi. [2]The carry out bits are generated by the carry look ahead logic.

3.2 Carry Skip Adder:

Because the Cin-to-Cout link is the longest in the ripple-carry-adder, it makes sense to try to speed up carry propagation through the adder. This is performed by employing carry-propagate pi signals inside the group [6]. If pi=1, the carry will bypass the whole group.
P(I,i+3)=Pi+3.Pi+2.Pi+1.Pi
The output of the AND gate is Ored with Ci+4 using the individual propagate values to obtain a stage output of

Carry = Ci+4 +P(i,i+3).Ci

If P(i,i+3)=0, the group's carry-out is determined by the value of Ci+4, as shown in the diagram. If P(i,i+3)=1 and the carry-in bit is Ci=1, group carry-in is automatically passed to the next group of adders. The term "carry-skip" refers to the fact that if the condition P(I,i+3) is met, the carry-in bit skips the block entirely. Ci is correct.

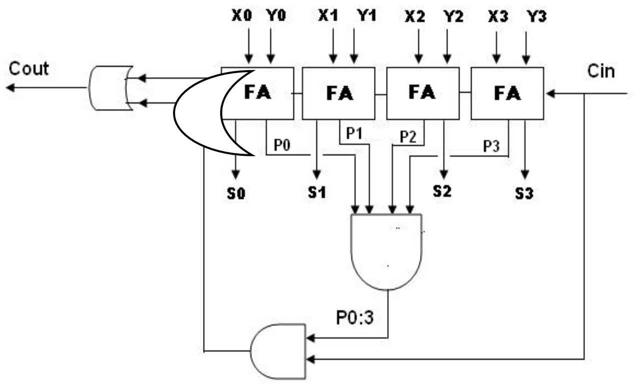

Figure 3.2: Carry Skip Adder

IV. CONCLUSION

We created the Verilog model of a 4-bit processor in this study and synthesised it in Xilinx ISE 8.1i. Ripple Carry Adder, Carry Look Ahead Adder, and Carry Skip Adders are just a few of the adders that have been created. When compared to the Ripple Carry Adder, the combinational delay path of the adders Carry Look Ahead and Carry Skip Adder is determined to be shorter. In Xilinx ISE 8.1i, the adders are also created. Higher order bit adders, such as 8-bit, 16-bit, 32-bit, and so on, have a greater decrease in delay route. The combinational delay route of the Carry Look Ahead Adder is greatly decreased since the amount of logic gates is lowered, according to the timing summary study of 8-bit adder units Ripple Carry Adder, Carry Look Ahead Adder, and Carry Skip Adder. Higher order bit adders, such as 8-bit, 16-bit, 32-bit, and so on, have a greater decrease in delay route.

REFERENCES

[1] Bruce Shriver and Bennett Smith, "*The Anatomy of a High- Performance Microprocessor*", IEEE Computer society Press, Los Alamitos, CA, 1998.

[2] James M. Feldman and Charles T. Retter, "*Computer Architecture*" McGraw-Hill, New York, 1994.

[3] Carl Hamarcher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", McGraw-Hill, New York, 2002

[4] BehroozParhami, "*Computer Arithmetic*", Oxford University Press, New York, 2000. A comprehensive, in depth treatment of the subject.

[5] Xilinx, Using Block RAM in Spartan-3 FPGAs, Dual-Port Block Memory Core v6.3,. www.xilinx.com.

[6] Prashant Gurjar, Rashmi Solanki, Pooja Kansliwal, "*VLSI implementation of adders for high speed ALU* " , International Journal of Computer Applications (0975 -8887) Volume 29 , No.10, September 2011.