Table of Contents

Section 1: Introduction

* Table of Contents
* Preface

Section 2: Usage of Processor

* Input Instruction Instructions
* Instruction Table

Section 3: Hardware Design

* I/O Interface Block Diagram
* Overall High-level Block Diagram

Preface

This document is meant for users, engineers, and internal reference. This document highlights the specifications, usage, and some level of the internal engineering (which is required / helpful for the usage of the processor).

This document does not describe the development or the details of the implementation of the product, such details can be found in the architecture documentation. This document merely describes the usage of, and the results produced by the processor in as accessible a way as possible.

Ultimately, however, it is imperative for anybody reading this document to keep in mind that this processor was created only for learning purposes, to enhance the knowledge of, and act as Verilog development practice for its creator. It is not meant or designed for any real-world applications or commercial use.

Input Instruction Instructions

The processor has a total of 5 inputs, they are – inputs, inpenable, cinputs, outenable, enable and reset. These serve the purpose of delivering control commands, enabling the inputting, delivering constant inputs, enabling the outputting, and resetting the processor respectively.

The reset is one bit wide and determines whether or not the processor is under reset. This reset is active high and will continue to keep the processor under reset for as long as it is high.

The inputs input consists of exactly 4 bits. The most significant three bits usually serve as the opcode (discussed in later chapters) and the least significant bit serves as a modifier to the instruction. The modifier can either state that the input of the instruction comes from the constant inputs or from one of the internal storage registors of the processor.

The cinputs input consists of a variable number of bits which is equal to the buswidth of the processor. This value determines the width of all data buses and therefore the maximum values that can be held and operated on by the processor and is determined before the synthesizing and implementation of the processor. The cinputs hold the constant inputs for the instructions that require them and that are modified to include them instead of the internal registors.

The inpenable and outenable inputs are both one bit wide each. They determine when the input is read, and the output is outputted. They both work on positive edges. To input one command, it’s control inputs and constant inputs must be placed in the inputs and cinputs respectively. Following this once the inenable bit goes from low to high, the input will be read and stored in the processor input memory. The outenable works in the same way except for extracting one output from the output memory on every positive edge.

The enable input is one bit wide. It determines when the processor reads the input memory, executes its commands and places the outputs in the output memory. It works on the positive edge and once it goes from low to high the processor processes all instructions in the processor’s input memory.

Instruction Table

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction | Assembly | Opcode | Description |
| Addition | ADD x | 1010 | Add from *x* to ACC |
| ADD REG x | 1011 | Add from reg *x* to ACC |
| Subtraction | SUB x | 1000 | Subtract *x* from ACC |
| SUB REG x | 1001 | Subtract reg *x* from ACC |
| Multiplication | MLT x | 0100 | Multiply *x* with ACC |
| MLT REG x | 0101 | Multiply reg *x* with ACC |
| And | AND x, y | 110x | And bit number *y* from ACC with *x* |
| Or | ORR x, y | 111x | Or bit number *y* from ACC with *x* |
| Load | LOD x | 0010 | Load *x* into ACC |
| LOD REG x | 0011 | Load reg *x* into ACC |
| Store | STR x, y | 0110 | Load *x* into reg *y* |
| STR x | 0111 | Load ACC into reg *x* |
| Clear | CLR | 0000 | Clear ACC and regs |
| Output | OUT | 0001 | Output ACC |

I/O Interface Block Diagram

Overall High-level Block Diagram