8-bit ALU

K Raja Pradyumna, P Vivek Vardhan

Indian Institute of Technology Hyderabad

EE5811 FPGA Lab

March 7, 2019

Overview

Project Outline

- Design an 8-bit ALU, also design a CPU along with it to interface the ALU.
- This interfacing CPU contains multiple general purpose registers, program counter, program memory, status register and an architecture similar to Intel 8080 CPU.
- Also, along with this CPU, a custom compiler will be designed to compile instructions into a binary format. The Binary encoding for these instructions follows the Intel 8080 Instruction set.
- Synthesize the module on the Icoboard and Interface it to analyze the outputs.

ALU implementation

- The ALU is implemented as an FSM.
- The inputs and outputs of the ALU are as follows:
  - Input - 1-bit clock
  - Input - 5-bit opcode
  - Input - 8-bit data operand A
  - Input - 8-bit data operand B
  - Input - 1-bit enable
  - Input - 1-bit input-ready
  - Input - 1-bit carry-in
  - Output - 8-bit result Y
  - Output - 1-bit result ready
  - Output - 1-bit carry-out
  - Output - 1-bit zero
  - Output - 1-bit negative
  - Output - 1-bit overflow
  - Output - 1-bit parity

ALU implementation

The list of operations performed by the ALU are as follows:

- ADD
- ADD WITH CARRY
- SUB
- SUB WITH BORROW
- NEGATE
- INCREMENT
- DECREMENT
- PASS THROUGH
- AND
- OR
- Exclusive-OR

ALU implementation

- One's COMPLIMENT
- Arithmetic SHIFT LEFT
- Arithmetic SHIFT RIGHT
- Logical SHIFT LEFT
- Logical SHIFT RIGHT
- ROTATE LEFT
- ROTATE RIGHT
- ROTATE Through carry LEFT
- ROTATE Through carry RIGHT

The CPU and the compiler

- The CPU design in this project is quite minimal and is primarily designed to demonstrate the ALU functioning.
- The CPU has seven general purpose registers, a status word, program counter and program memory.
- The CPU also supports branching instructions similar to an **if statement**.
- The code in the program memory is loaded by auto-generating the verilog code in python.
- The CPU is implemented as a state machine with the states being :
  - Instruction fetch
  - Instruction decode
  - Execute
  - Data write-back
  - Program counter update

Interfacing the ICOboard

- The current plan is to interface the CPU with **arduino** using the ICOboard pinout.
- If the time permits, the plan is to use a **16 by 2 LCD display** to display the results.

Thank You