Zakary Holliman
385 Directed Study
Dr.Franchi
5/19/2022

Basic Manual and Schematic of CPU

CPU Schematic

The CPU is comprised of 6 main parts

- Clock
- Program Counter
- Program Memory
- Bus
- RAM
- ALU

Clock

The Clock is the central part of the CPU that keeps everything synchronous. It uses a basic Logisim clock but also has some inbuilt functionality to allow turning the clock on and off, and stepping through one cycle at a time for debugging. It also includes a Halt which can be programmed to trigger and stop the clock as well.

The Clock runs alongside the main bus, and is a direct input into every other part of the computer, keeping everything synchronous.

Program Counter

The Program Counter is a 4 bit counter that continually counts from 0 to 15, which is used to address instructions in the Program Memory. It counts using JK Flip Flops with the Clock as an original input, and each Q thereafter being used as an input to the next JK Flip Flop. This essentially divides the Clock in half for each Flip Flop, which results in a counting pattern where we see the address count from 0 to 15, and then reset.

The Program Counter has an Enable, which is used to determine whether it should be counting or not, as well as a Jump Address and Jump Enable. The Jump Address can be used to set the Program Counter to a specific value, which is used in things such as loops or system resets.

Program Memory

The Program Memory houses all of the instructions that are fed out into the rest of the CPU. It has 16 address lines and thus can hold 16 different instructions.

The Instruction Addresses are read in from the Program Counter and put into a 4 bit decoder, which allows for 16 different instruction addresses.

Inside the Program Memory is a Data line, and two Instruction Lines.

The Data Line allows a value to be output onto the Bus of the CPU

Instruction Lines 1 and 2 allow instructions to be decoded and executed by the CPU.

Bus

The Bus is a central data line that Program Memory, RAM, and ALU are all connected to. It allows one part of the CPU to talk to any other part of the CPU, and every register is connected to it.

ALU

The ALU is the brain of the CPU, and performs all of the arithmetic needed.

It has two Registers, Register A and Register B. These Registers are connected to the Bus and can read values from it to be used in computations.

The ALU automatically performs operations using Registers A and B, and only needs to be output to the Bus using the Tri-State Enable if the result wants to be read.

The ALU can Add or Subtract. Addition is the default operation, but the Subtract option can also be selected. This will invert every bit of Register B, and feed a 1 in as the Carry Bit of the Adder used by the ALU. Due to how Two's Compliment numbers work, this will result in Subtraction.

RAM

Lastly is the RAM, or Random Access Memory. The RAM is temporary memory that is stored while the CPU is in use.

The RAM is comprised of 16 8-bit registers, all connected in series. It reads in and out to the Bus using Tri-State logic.

It has two address lines, one for Reading and one for Writing.

The Read address line will read either from the Bus (address 0) or from a specific register in the RAM at the specified address line. It will read this value out onto the Bus, which allows it to be accessed either in another part of the CPU such as the ALU or in another register of the RAM.

Starting CPU

To first start the CPU you will want to go to Simulate and make sure “Simulation Enabled” and “Ticks Enabled” are both set to true.

You will then click the “Enable Clock” button, which will start pulsing the clock, if everything is working correctly, it should be ticking at a steady rate out onto the bus, and the “CLK” LED should be flashing. To properly observe the program running without it being too fast, the tick rate should be set to 2Hz or 1Hz.

After that, you can press “Enable” on the Program Counter. This will begin counting up from 0 to 15 and addressing the hardcoded instructions on the Program Memory to execute the program.

Program

The program that comes loaded onto the CPU is a very basic program that has 6 instructions.

There are only 4 instructions that are built into the CPU currently

- 0 - Halt Program - Triggers the Halt on the Program Counter, causing it to stop counting.
- 1 - Write ALU A - Enables the Write on ALU Register A, which will write whatever data is currently on the Bus to the register on the next Clock cycle
- 2 - Write ALU B - Enables the Write on ALU Register B, which will write whatever data is currently on the Bus to the register on the next Clock cycle
- 3 - Enable ALU to Bus - Enables the ALU output to Bus, which reads the current value of the ALU’s operation to the Bus.

Instruction 1

- The first piece of this instruction is a Read Only Memory (ROM) that has the value of 1, which means it is the Write ALU A instruction. The second block is a Read Only Data (ROD) with a value of 5. This instruction will output a data value of 5 onto the RAM, and write that data to ALU Register A.

Instruction 2

- This instruction is a ROM with a value of 2 (Write ALU B) and ROD with value of 1. This means that it will write a value of 1 to the Bus, and write from the Bus to ALU Register B, meaning 1 will be written to ALU Register B.

Instruction 3

- This instruction is two ROMs, the first has the value of 3 (Enable ALU to Bus) and the second is 1 (Write ALU A). This means that the computed value of the ALU will be output onto the Bus, and immediately be read in and written to Register A of the ALU again.

Instructions 4 and 5

- These instructions are repeats of Instruction 3, and basically increment the value of Register A by 1 on each interaction

Instruction 6

- This instruction is two ROMs, the first with a value of 0 (Halt Program) and the other with a value of 3 (Output ALU to Bus). This means that it will output the value of the ALU to the Bus to be read, and halt the program counter, resulting in the computed value of the loop to be shown to the Bus.

So essentially, you can see that this program reads in two values to begin, 5 and 1, and continually increments the result by 1 three times, and then displays the result and stops, displaying 9.