**Student Name: Rama Ibseileh**

**Date:10/21/25**

**Introduction:**

**This project gave me hands on practice with two of the most important parts of a CPU registers and memory. Using the Digital simulator, I designed several 4-bit registers, a program counter, an instruction register, and a simple DRAM module. Then I connected them together to show how data moves inside a basic processor. The goal was to understand how instructions are stored, fetched, and executed in hardware**

**PC**

**The program counter keeps track of the memory address of the next instruction. I built it as a 4-bit register connected to a small adder that adds 1 to the current value. A multiplexer decides whether the PC should increment normally or load a new address.**

**Control signals used:**

* **INC: increments the PC value (PC + 1).**
* **LD: loads a custom address.**
* **RST: resets the PC back to 0000.**

A diagram of a computer

AI-generated content may be incorrect.

A diagram of a circuit

AI-generated content may be incorrect.

**IR**

**The instruction register holds the instruction currently being executed. It’s another 4-bit register that loads data from the memory bus when LD is active. This register makes it possible to fetch an instruction from memory and hold it long enough for decoding and execution.**

A diagram of a circuit

AI-generated content may be incorrect.

A diagram of a circuit

AI-generated content may be incorrect.

**GPR:**

**The GPR module has four 4-bit registers labeled R0, R1, R2, and R3. These act as temporary storage for data and results from computations.  
Each register can be loaded, stored, or cleared using control signals:**

* **Load: puts a new 4-bit value into the selected register.**
* **Store: sends the value out onto the data bus.**
* **Clear: resets the register to 0000.**

**I used a multiplexer to choose which register to load and tri-state buffers so that only one register drives the bus at a time.**

**SS**

A computer screen shot of a diagram

AI-generated content may be incorrect.

A diagram of a computer

AI-generated content may be incorrect.

**DRAM**

**For memory, I created a 4-bit wide, 16-address DRAM. It has:**

* **Address selects one of the 16 memory locations.**
* **Data the data bus.**
* **R/W controls whether we read or write (1 = Write, 0 = Read).**
* **CS chip select; must be 1 for any operation.**

**When writing, data is placed on the bus and stored at the chosen address. When reading, the data from tha address appears on the output.**

**A computer screen shot of a computer

AI-generated content may be incorrect.**

**A computer screen shot of a computer program

AI-generated content may be incorrect.**