SOUTHEAST EUROPEAN UNIVERSITY

FACULTY OF CONTEMPORARY SCIENCE AND TECHONLOGY

STUDY PROGRAM: “COMPUTER ENGINEERING”



SEMINAR PAPER FROM THE SUBJECT: EMBEDED MICROPROCESSOR SYSTEMS

THEME: “Z-80 Computer”

SUBJECT LECTURE: STUDENT:

PROF. DR. BESNIK SELIMI IRFAN URUÇI

TETOV, JANUARY 2025

Table of Contents

[Overview 3](#_Toc188348112)

[Z-80 Overview and architecture 3](#_Toc188348113)

[Adders 4](#_Toc188348114)

[Control logic and Instruction decoding system 7](#_Toc188348115)

[Circuit design 7](#_Toc188348116)

[Registers 8](#_Toc188348117)

[8 Bit Registers 9](#_Toc188348118)

[16-bit Registers 13](#_Toc188348119)

[Flags 14](#_Toc188348120)

[ALU 15](#_Toc188348121)

[Data Bus and Address Bus 17](#_Toc188348122)

[Data IN/OUT and Control Signal Integration 18](#_Toc188348123)

[CPU 20](#_Toc188348124)

[Computer with the custom Processor and RAM 25](#_Toc188348125)

[Instructions 26](#_Toc188348126)

[Conclusion 27](#_Toc188348127)

[References 27](#_Toc188348128)

# Overview

In this project , I am developing a computer in Logisim based on the Z80 architecture originally developed by Zilog company. The computer will be based on the Z80 architecture with some modification like a modified control logic and instruction decoding system , instructions and a slightly modified system overall. My goal is not only to gain practical experience in computer architecture and logic design in a hands-on project but to continue building on my knowledge in computer engineering and computer hardware design but also, I want to achieve a custom Z80 computer that could help in modern embedded systems.

# Z-80 Overview and architecture

The Zilog Z80 is an 8-bit microprocessor introduced in 1976 by Zilog, it was widely used when launched in personal computers but also in embedded systems and gaming consoles later on. The Z80 architecture features a 8-bit data and has a 8-bit data bus, and a 16-bit address bus that allows it access to 64 KB of memory, it typically operates at 2.5 MHz up to 20 MHz. It includes GPR (General Purpose Registers) with a mix of 8-bit registers (B,C,D,E,H and L).

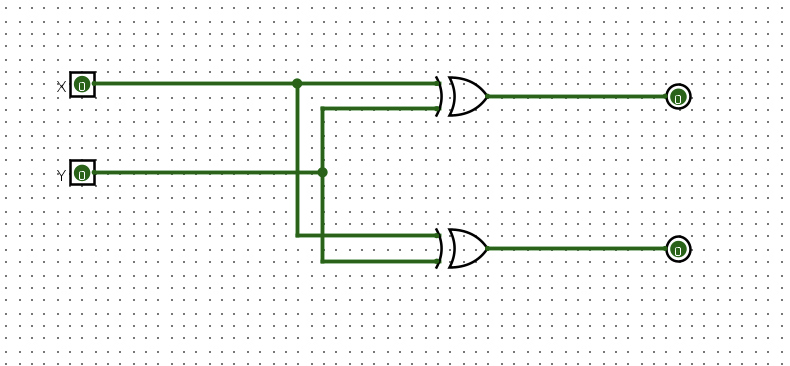
Also, Z80 contains a accumulator(A), Flag register(F) an alternate register set with a duplicate of GPR and a accumulator/flag register for faster context switching. It also had specialized 16-bit registers IX and IY but also a 16-bit PC (Program Counter ) and a 16-bit Stack Pointer (SP) an Interrupt Vector(I) and Memory Refresh(R) registers.

It has a instruction set that features over 150 instructions supporting from arithmetic and logical operations to Bit manipulation, Data transfer instructions, but it also has a backward compatibility with Intel 8080.

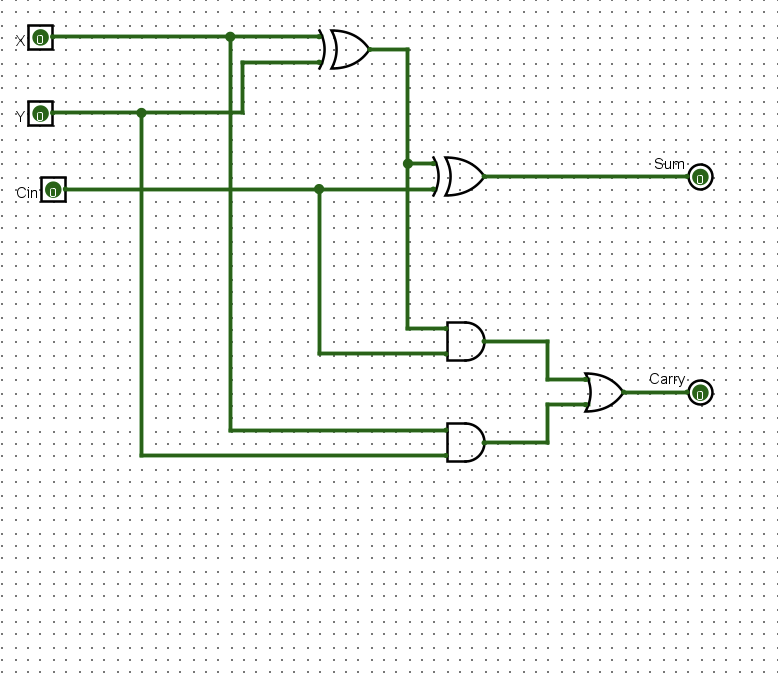
It still being used industrial controllers like in factories or production lines or even products like calculators and basic control systems still rely on the Z80-based controllers but also its still being used in some older systems in aerospace and military and healthcare equipment due to its stability and low-cost maintenance and the Z80 is still being maintained and updated by Zilog.

# Adders

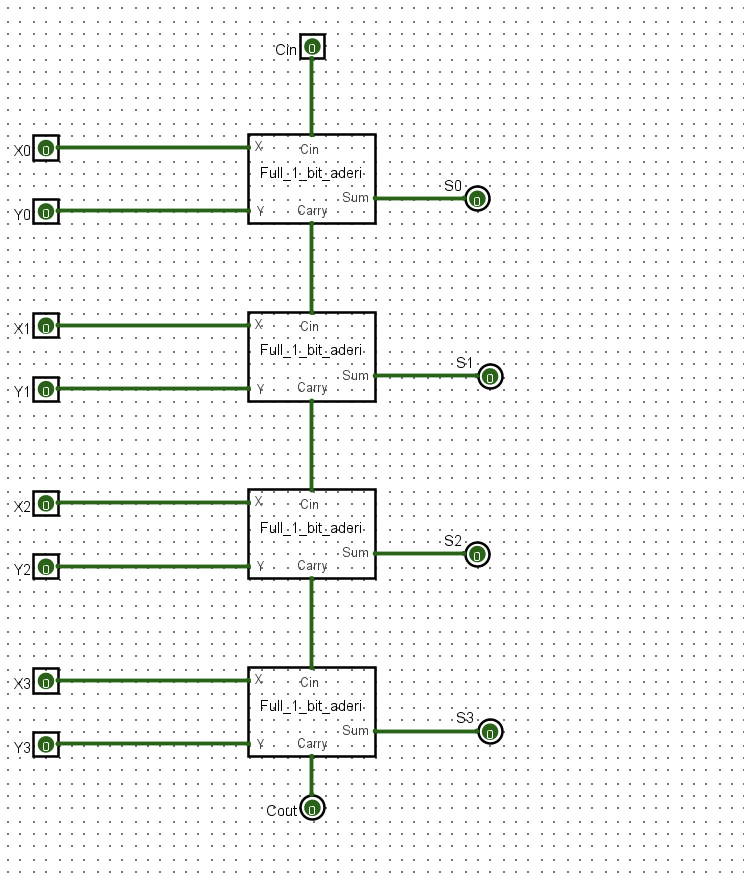
For Half adder in the project named “Half-Adderi” is the simplest form for adders :



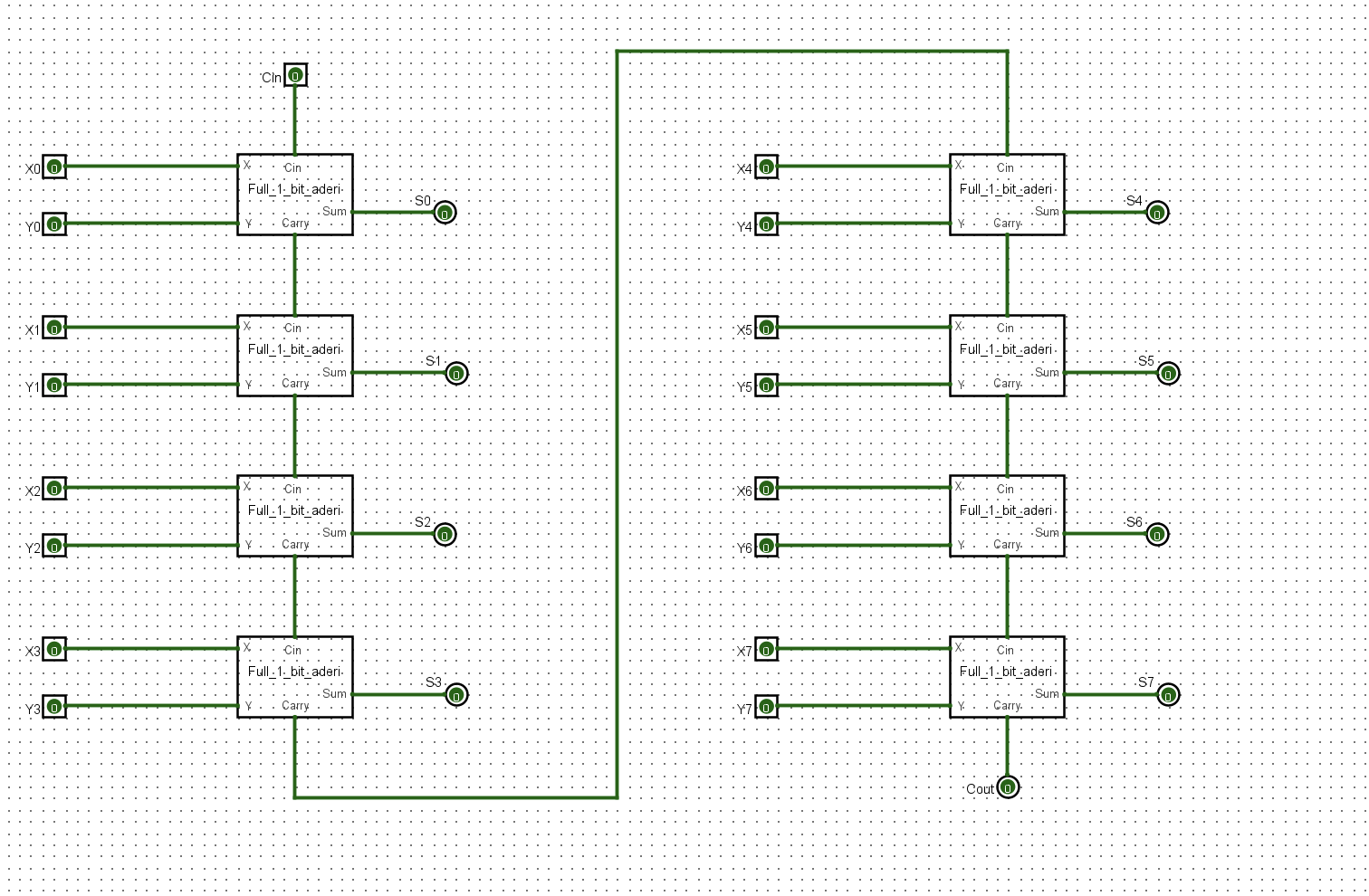
After that a full 1-bit adder adds three bits where 2 are input bits and one is a carry on bit:



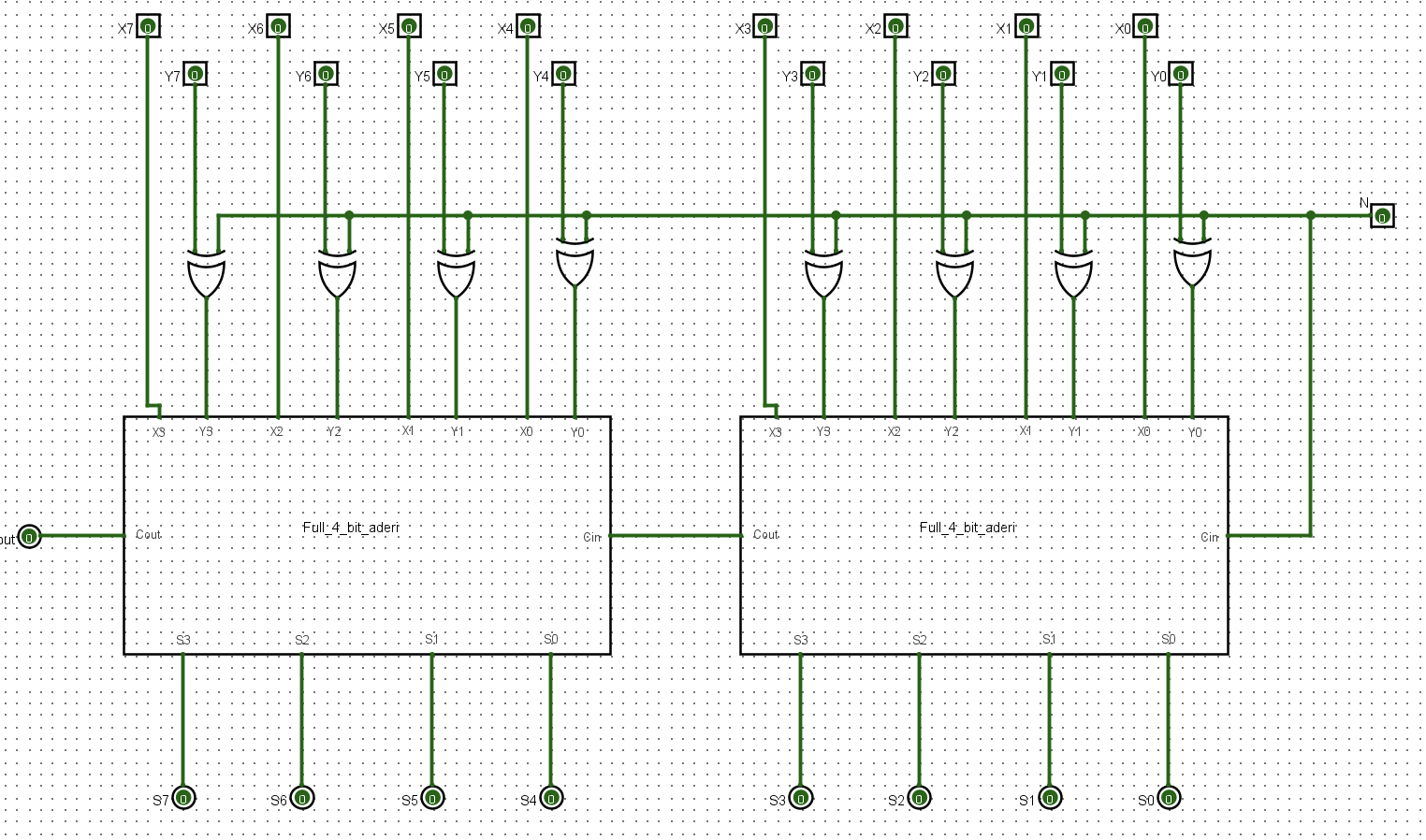
After implementing a 1-bit full adder I created a 4-bit full adder consisting of four 1-bit full adders



Similarly consisting from eight 1-bit full adders a 8-bit full adder:



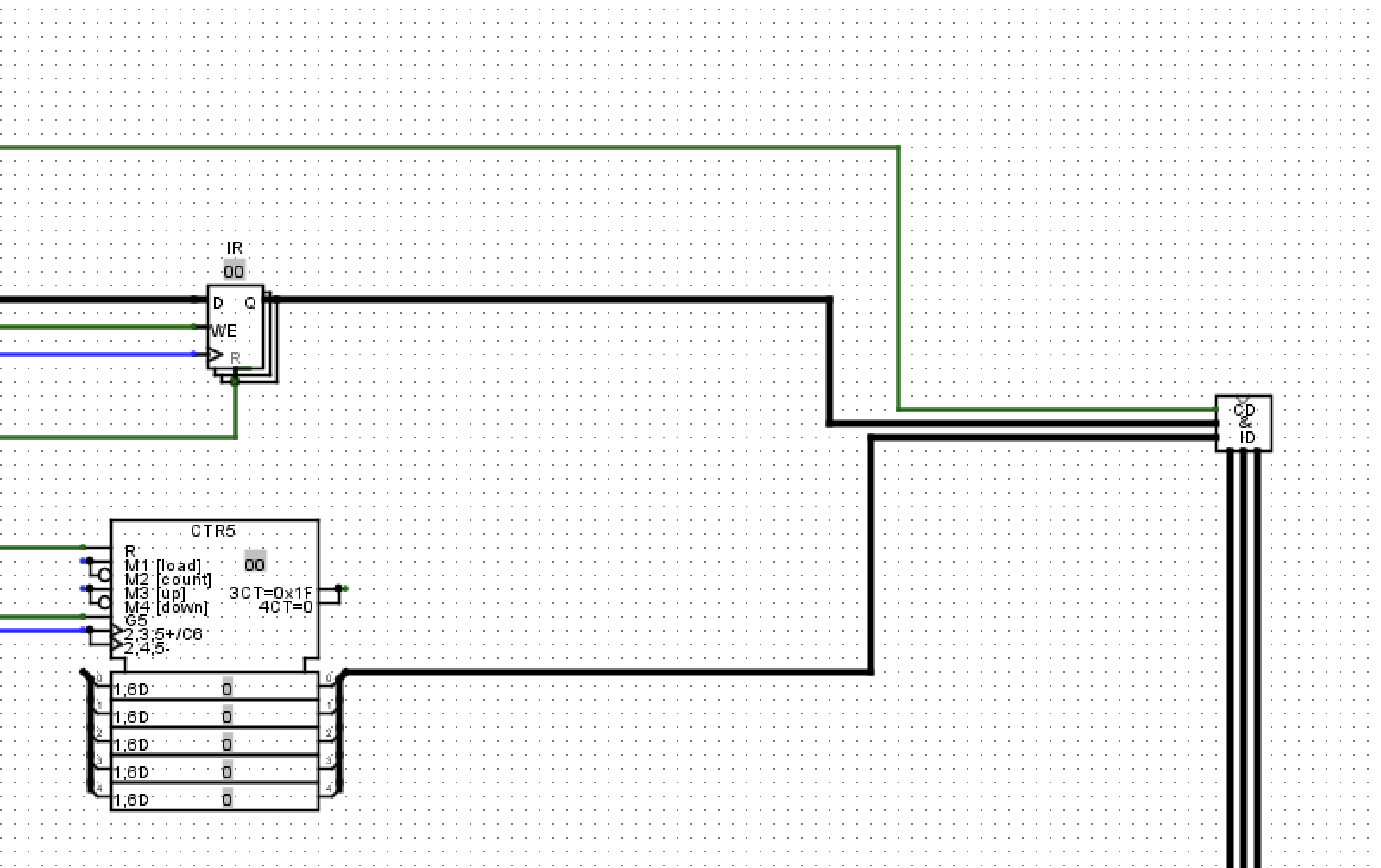
Using two 4-bit full adders created also a 8-bit full adder and subtractor



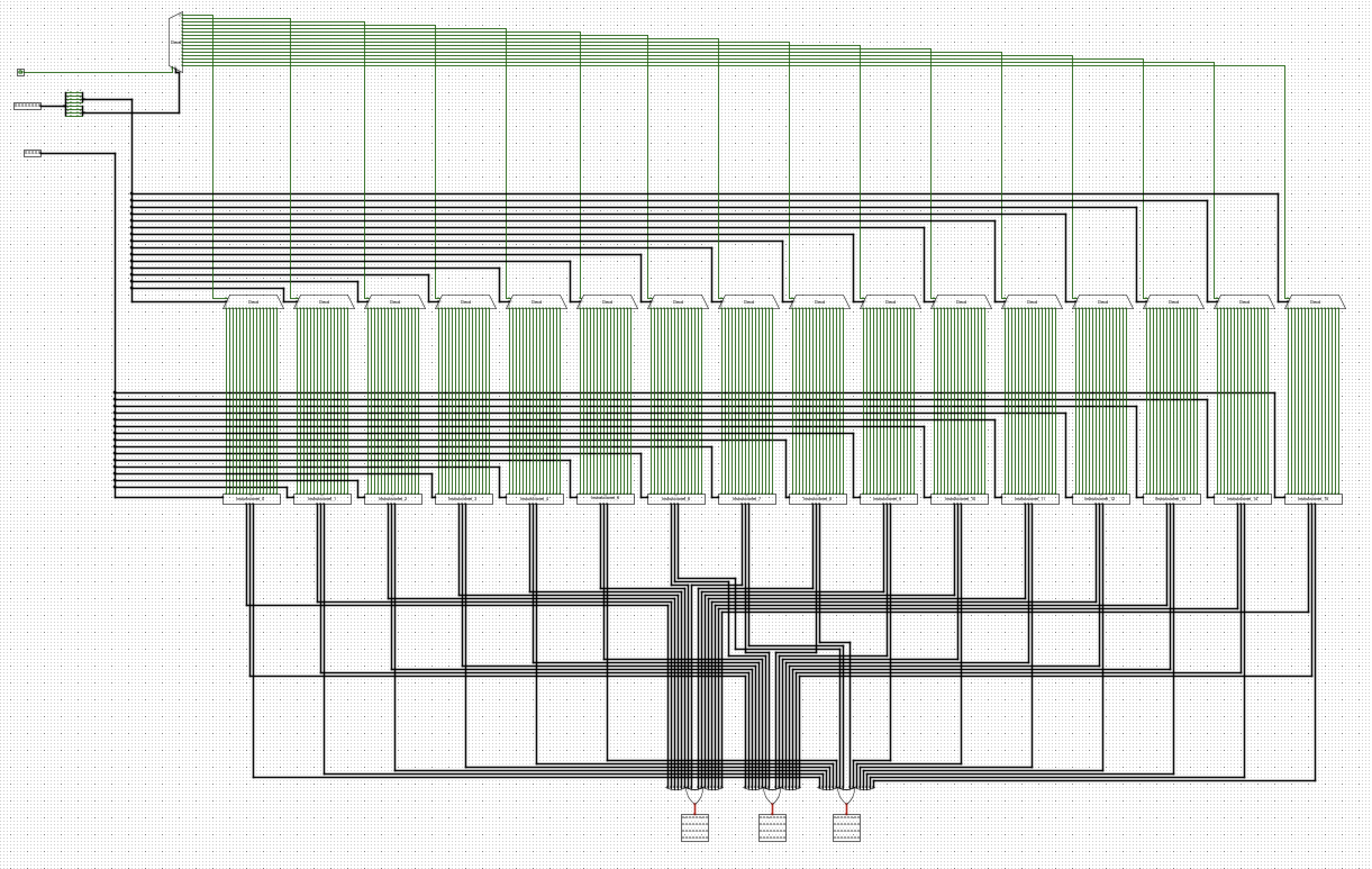
# **Control logic** and **Instruction decoding system**

In the traditional Z80 microprocessor it has a straightforward instruction decoding mechanism and control logic that directly translates the opcodes into executable actions on my design My design have changed with more advanced decoding capabilities but also a broader range of instructions for more complex control logic that can handle more operations simultaneously to reduce the latency. Where with the enhancements it has additional logic units to manage more complex operations but also new functionalities not available in Z80 design.

## Circuit design



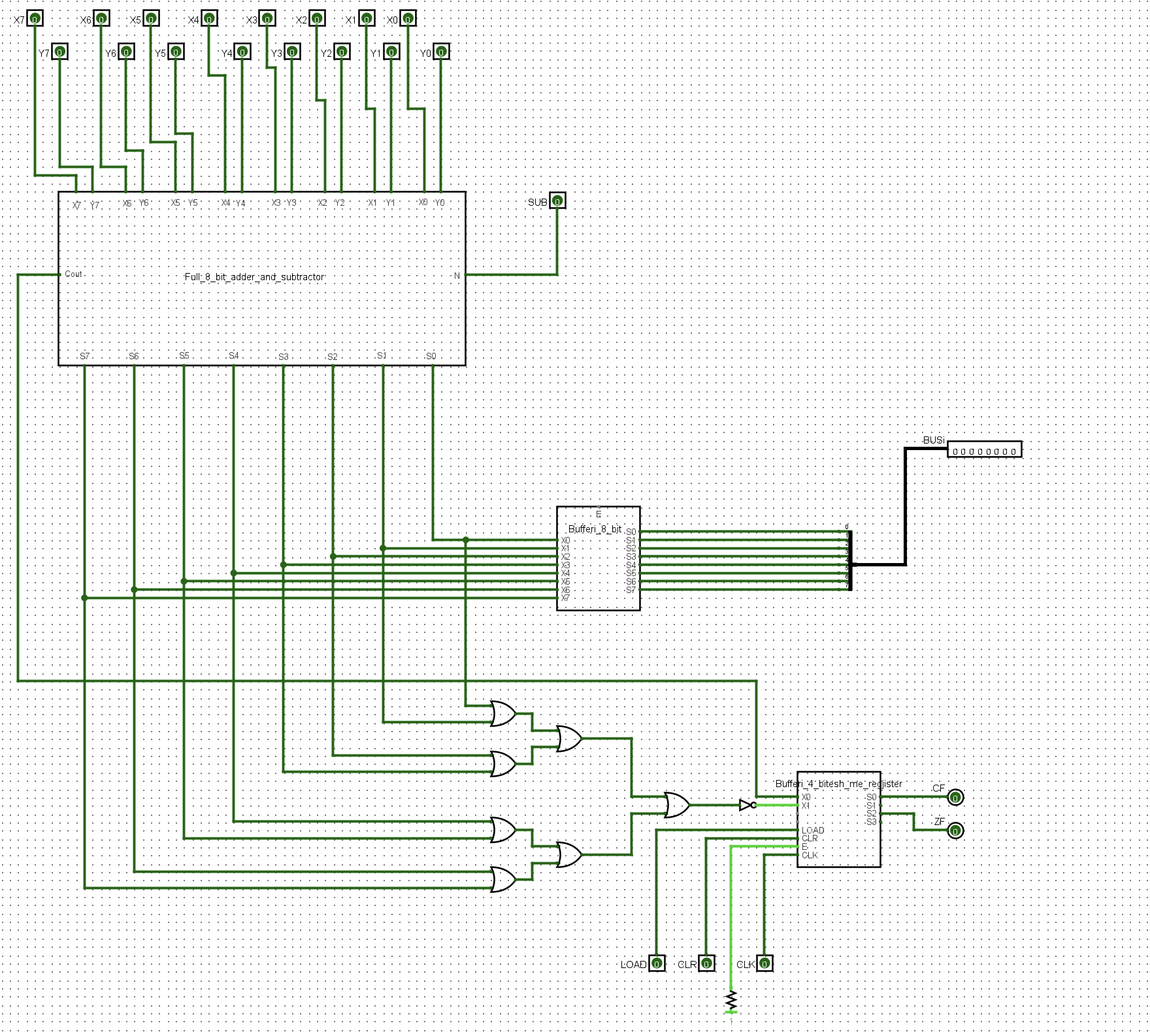
As mentioned before with this control logic and instruction decoding system, I have added more advanced decoding capabilities but also a wider range of instructions. The circuit above is made by an Instruction Register (IR) and a Controller. After that the main logic or CD & ID is made by decoders and a range of Instructions from 0x to 15x.



# Registers

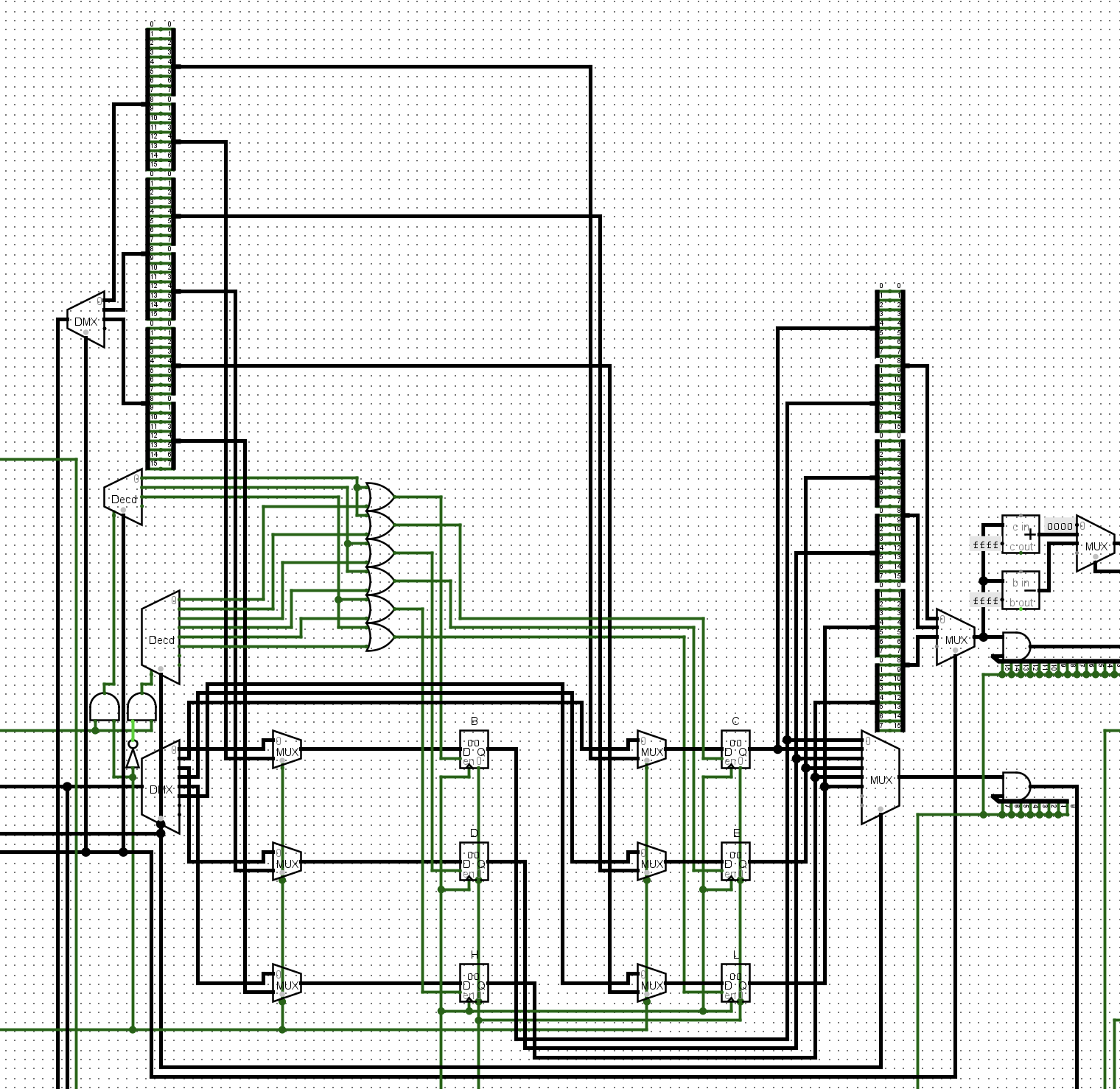
There are different registers in Z80 CPU some are 8 bit and some are 16-bit, where 8 bit are registers like B,C,D,E,H and L. While for 16 bit registers on the Z80 there are registers like PC , IX,IY and SP .But also there is the Instruction Register(IR), Flag register(F) ,TEMP and A that are more specialized registers.

There is a register I used for testing that used a 8-bit full adder and subtractor:

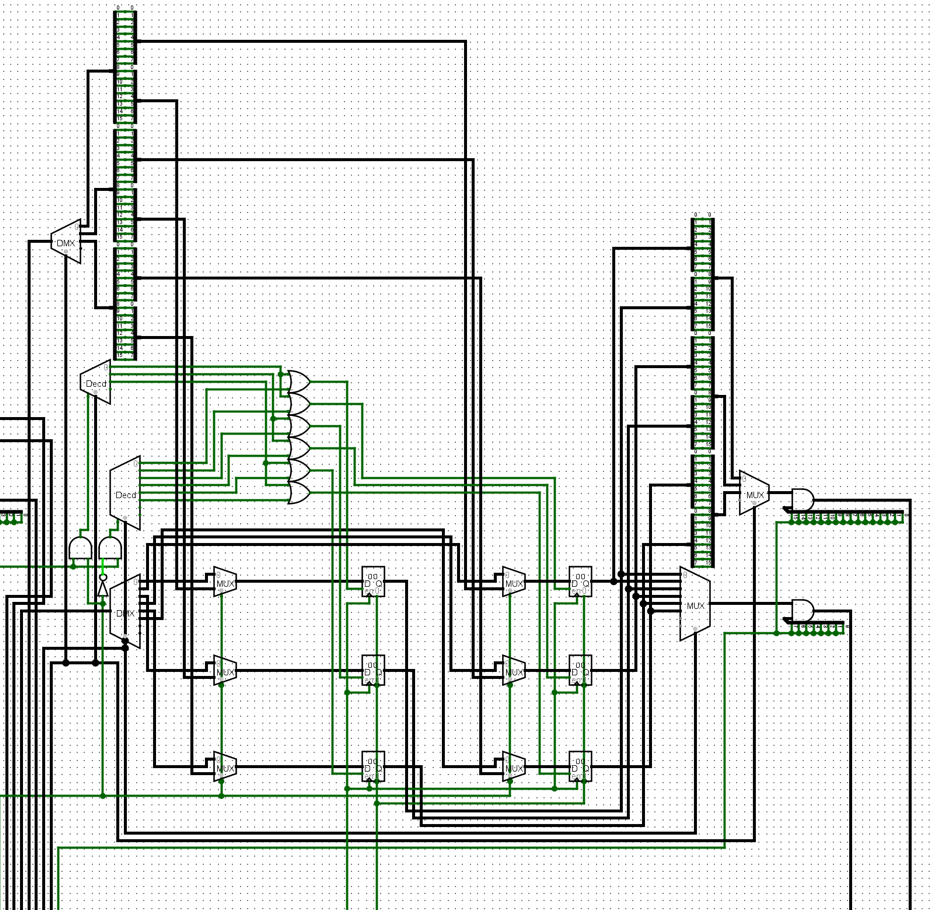


8 Bit Registers

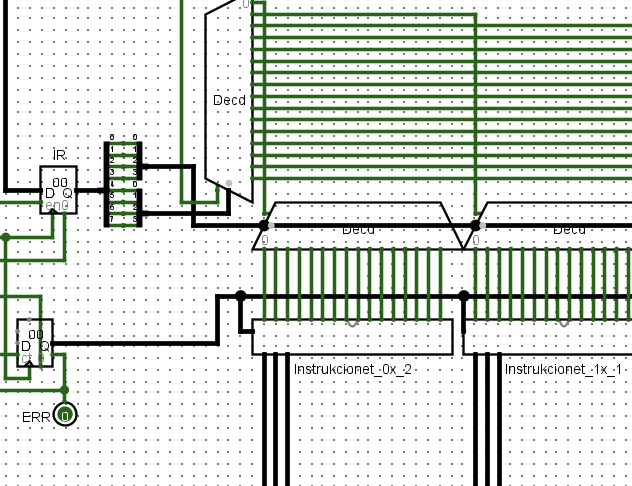
In Z80 architecture there are 8bit registers for general purpose like B,C,D,E,H and L



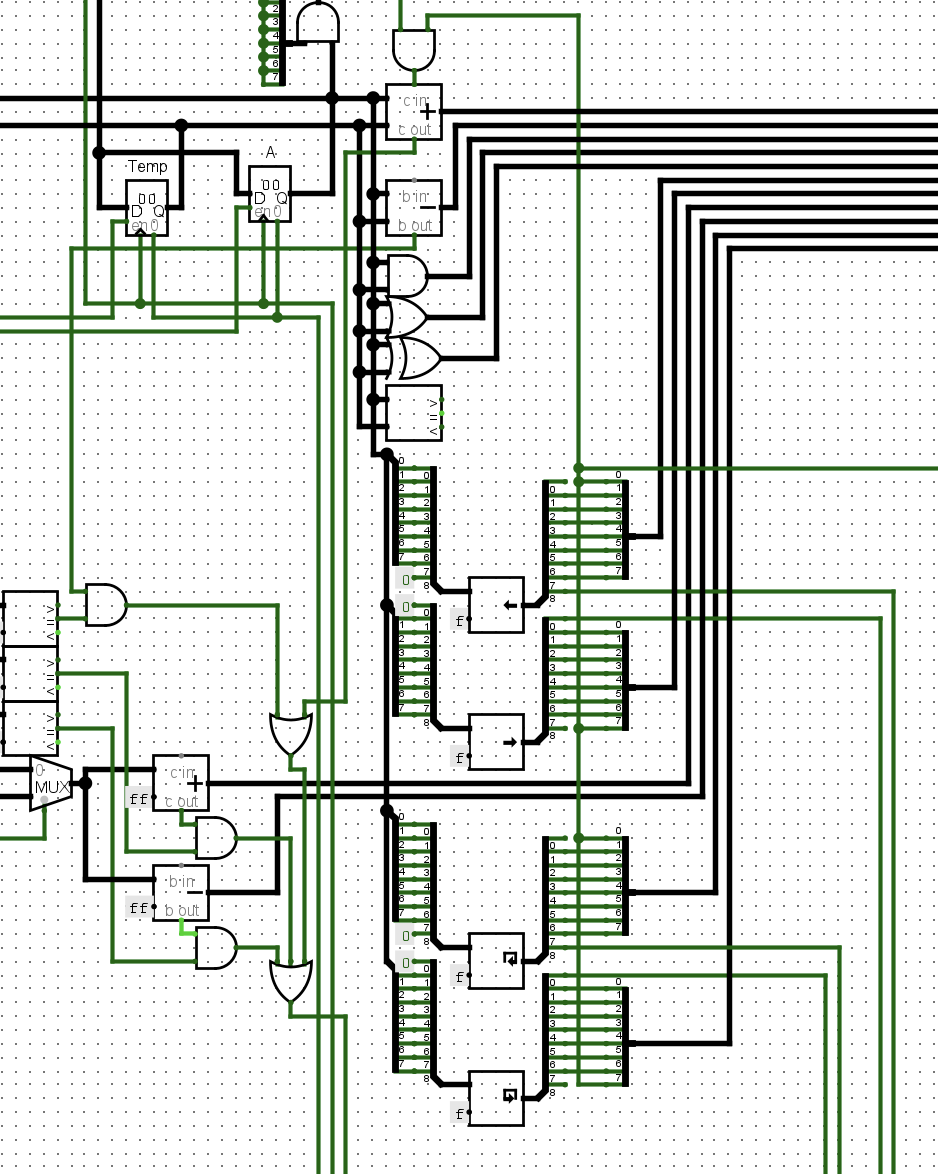
Also there is registers like B’,C’,D’,E’,H’ and L’



And also IR register used on the instruction decoding and control system:



TEMP and A registers are registers used closer to ALU used for arithmetic, logical and data transfer operations:



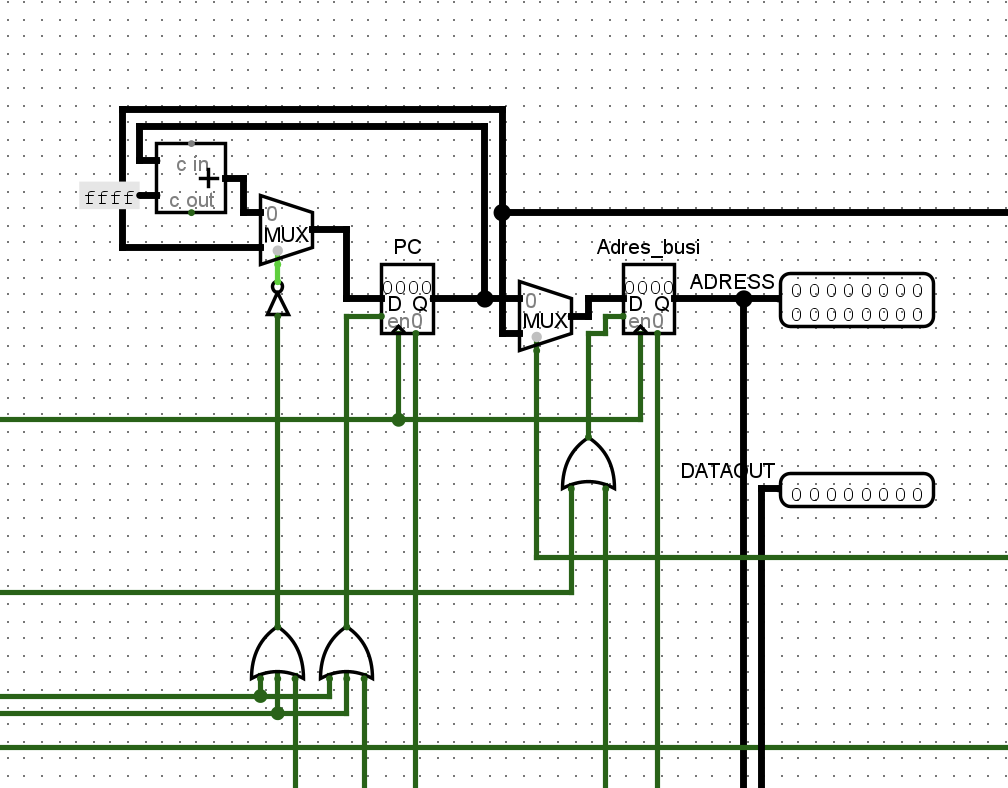
There is also the F register that is the register for storing the status of the flags used for operations performed by the processor:

A diagram of a circuit

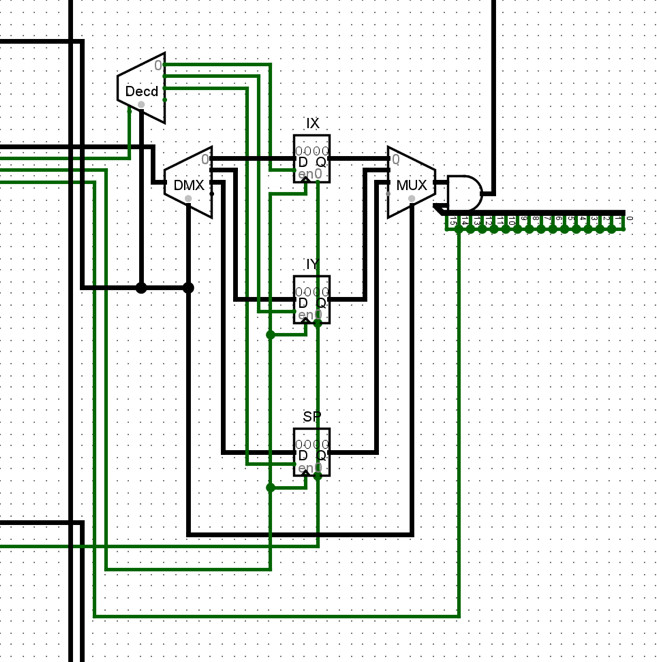
Description automatically generated

## 16-bit Registers

For 16-bit registers this are more specialized registers.   
One specialized 16-bit register is Program Counter (PC):



There are other 16-bit specialized registers like IX,IY,SP:



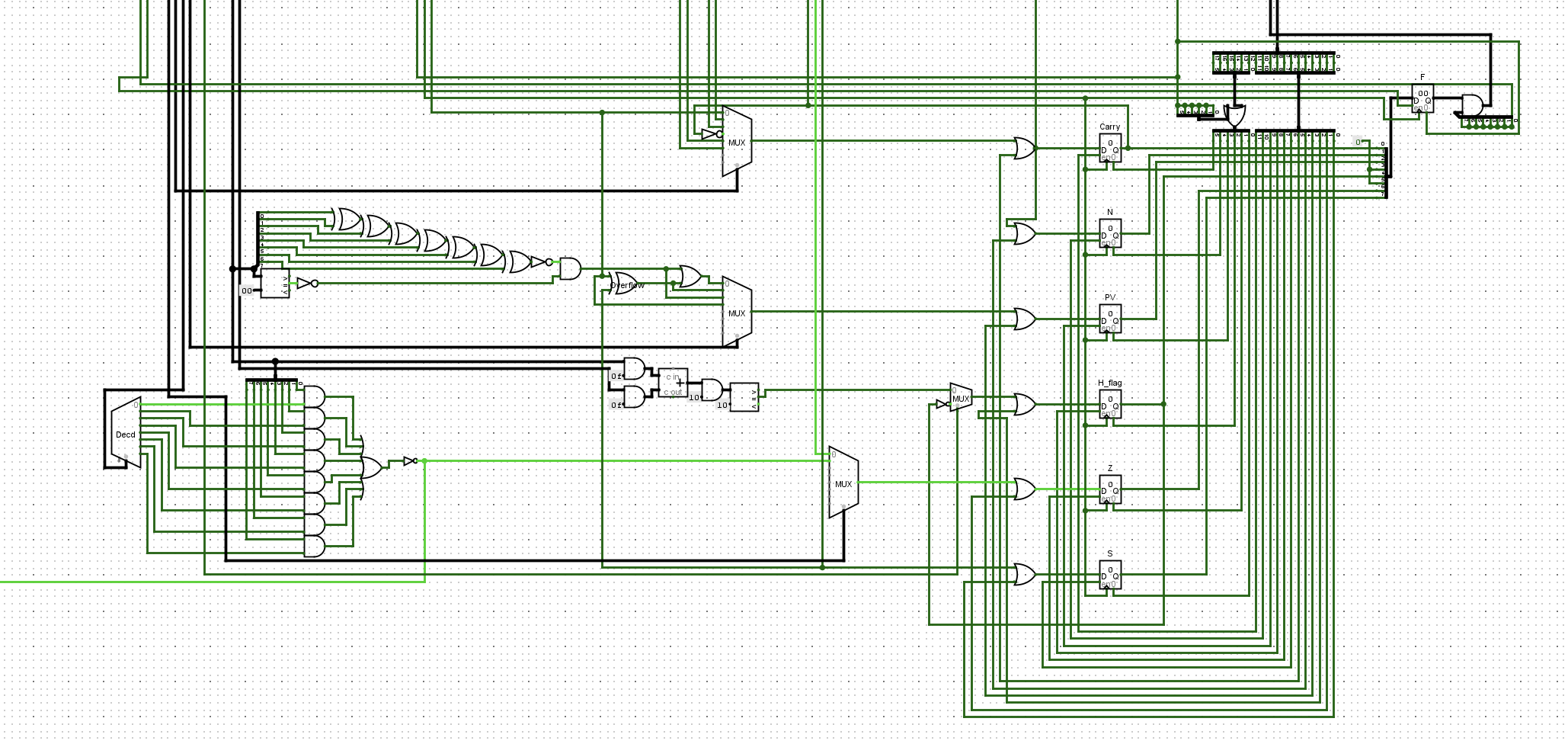
# Flags

Flags on the Z80 architecture like in other computers are important for telling the next operation and there are flags like Carry flag ,Subtract flag(N) , Parity or overflow (PV) flag , the Half Carry Flag (H) , the Zero (Z) flag and the Sign (S) Flag:

A diagram of a circuit

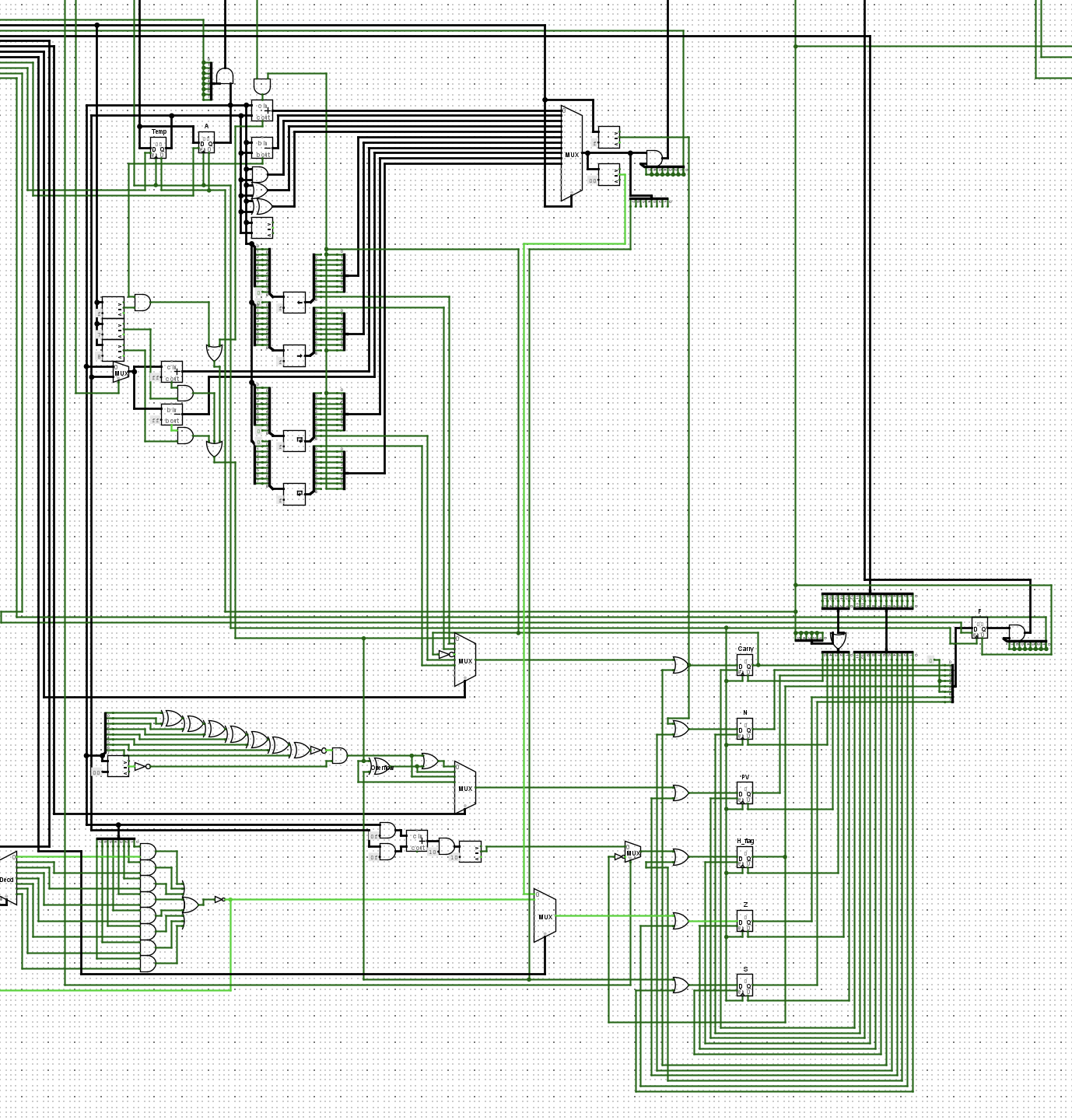
Description automatically generated

After this I connected the Flags with the Flag register to use for the processor operations , I used also 11 AND gates , 8 XOR Gates , a comparator , 3 Multiplexers , a adder , decoder , 7 OR gates and splitters accordingly :



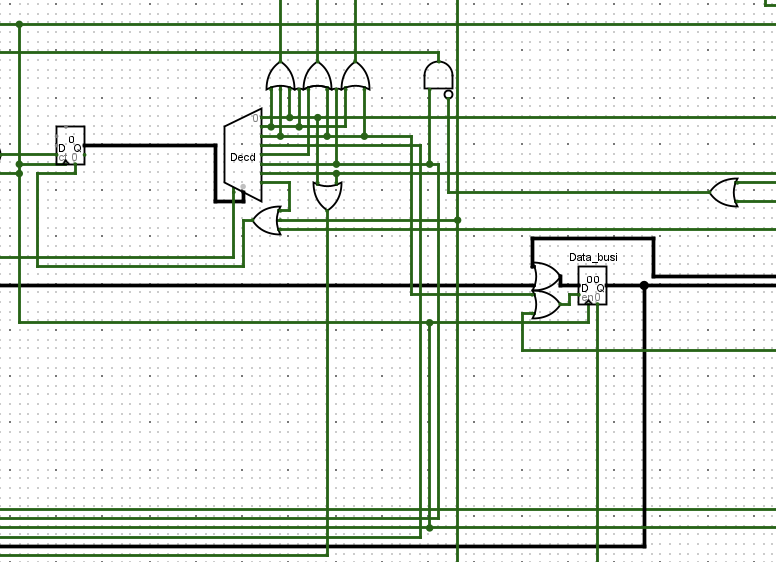
# ALU

ALU is a critical part of the CPU and in this project using the z80 architecture I designed a ALU where it’s going to use TEMP and Accumulator (A) specialized registers as mentioned in the 8-bit registers section for holding the data that is processed at that time in the CPU close to ALU. The ALU uses also the Flags explained previously and with this two parts close to ALU I build the ALU with 2 shifters, one that shifts the bits logically left and the other logically right , also two other shifters where one rotate bits right and one shifter that rotates them left .In the ALU there are 6 comparators , two adders and two subtractors. The final ALU :

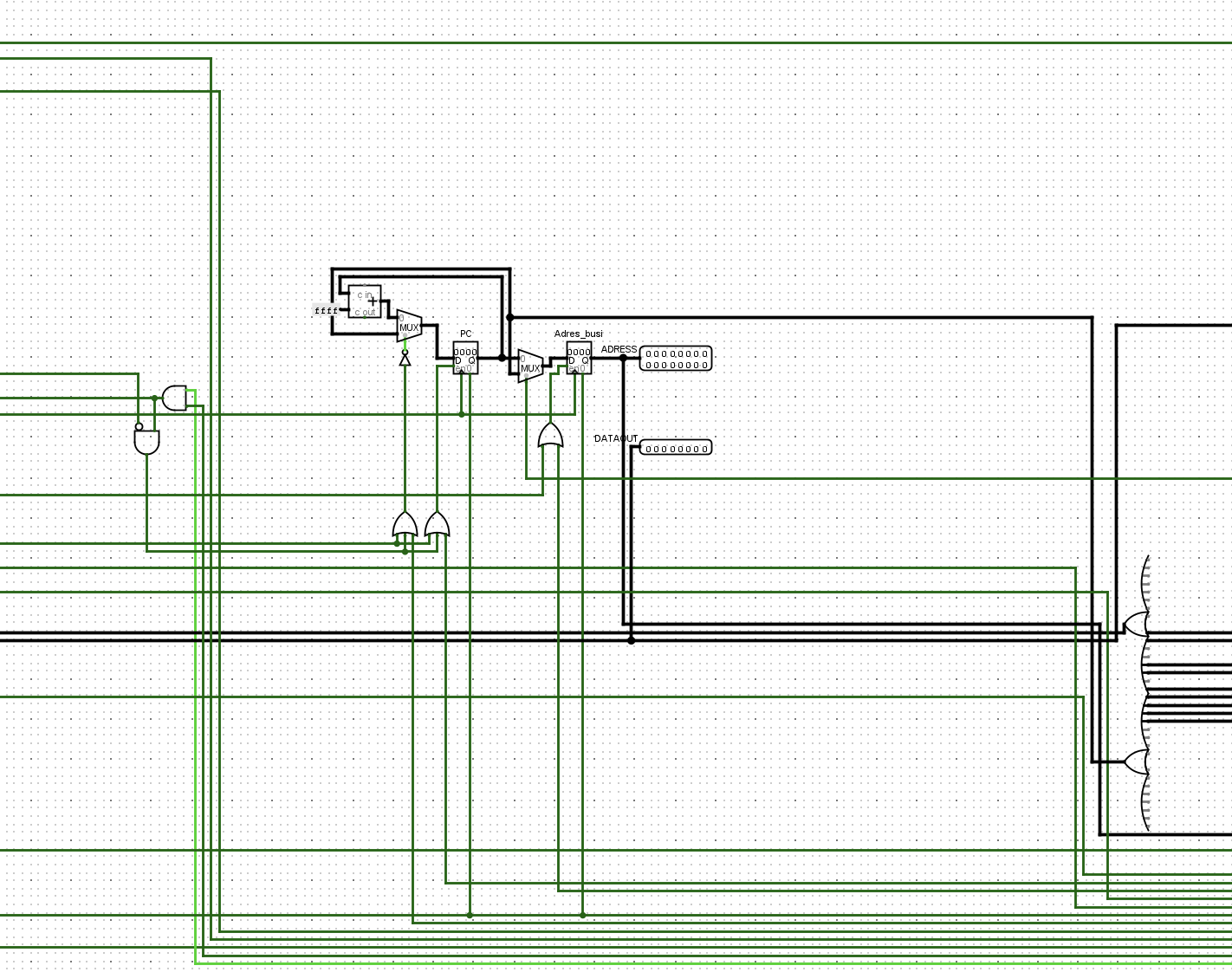


# Data Bus and Address Bus

Z80 utilizes two primary buses for managing data flow and memory addressing in the system, the data bus and address bus. The Data bus is a 8bit bus for transmitting the actual data between the different parts of the CPU and the computer. While the Address bus its 16 bit that consists of 16 lines and can address 2^16 which is 65536 memory locations directly its used exclusively by CPU for specifying the memory address where the data is going to be read or written. Unlike the data bus the Address bus is unidirectional .  
  
The Data bus in the project:

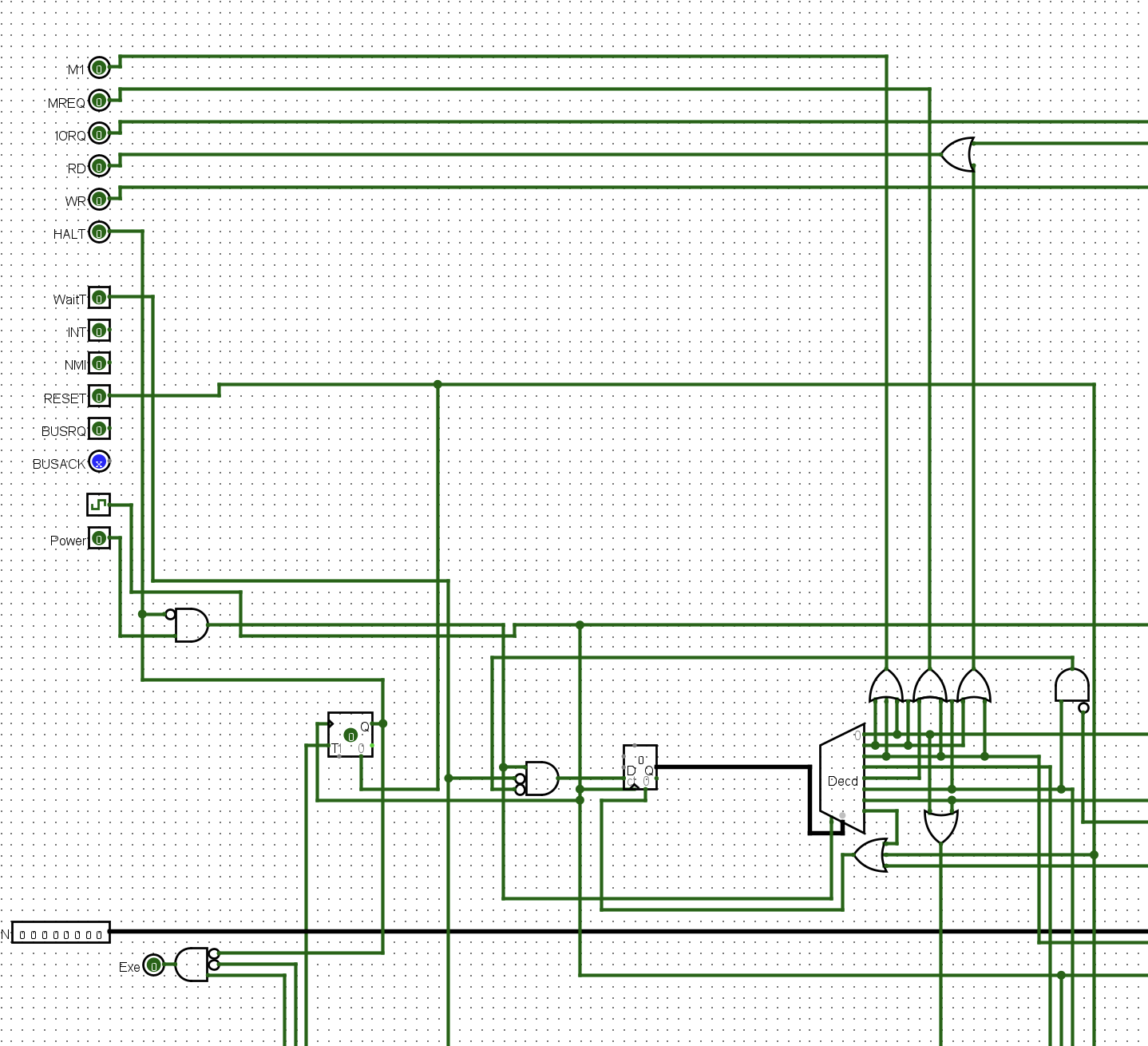


Address Bus in the project:



# Data IN/OUT and Control Signal Integration

The Data In is where data in a 8-bit format enters the CPU from RAM , while the Control Signals like Clock for example control the flow of the data and their execution time

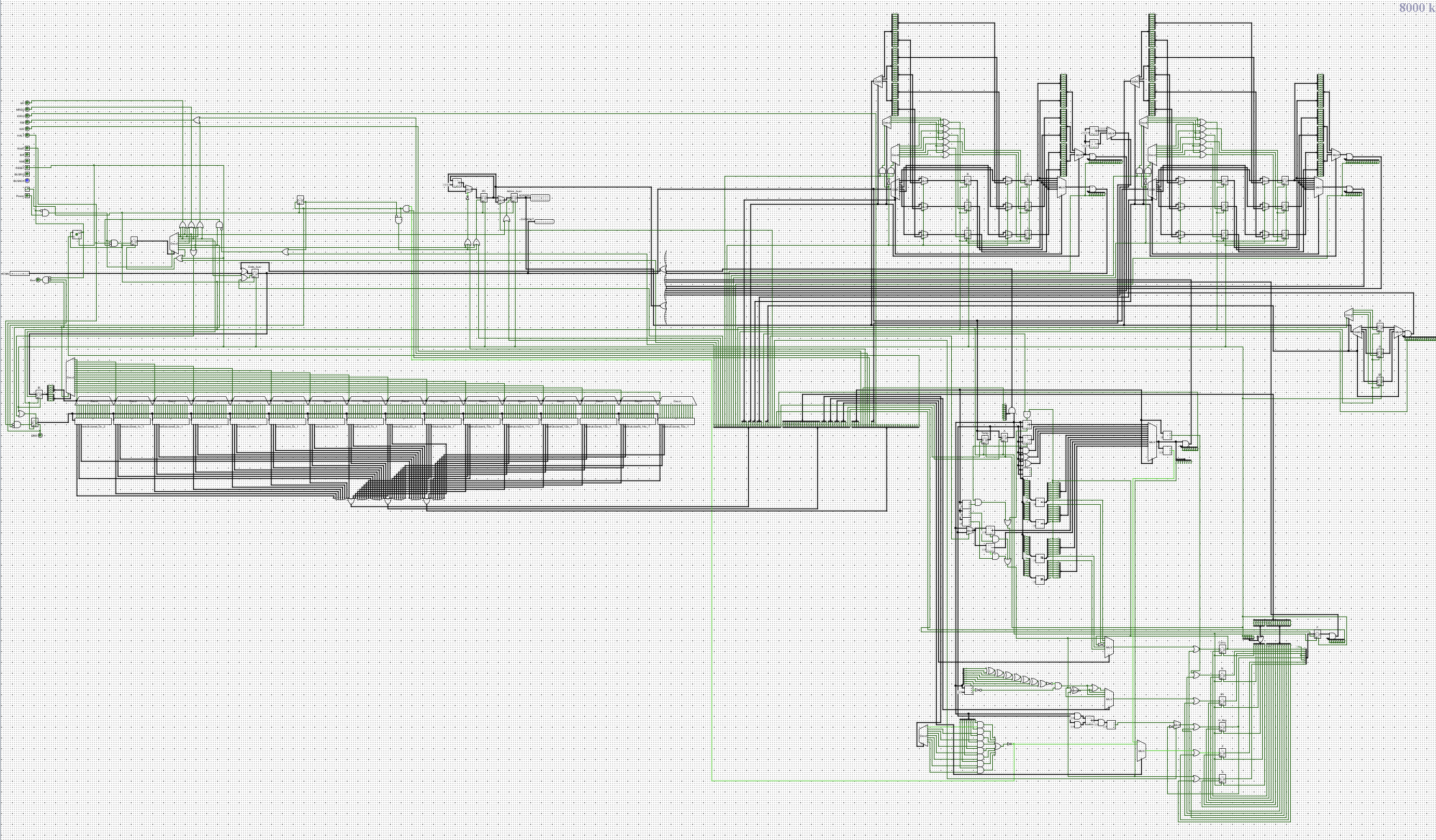


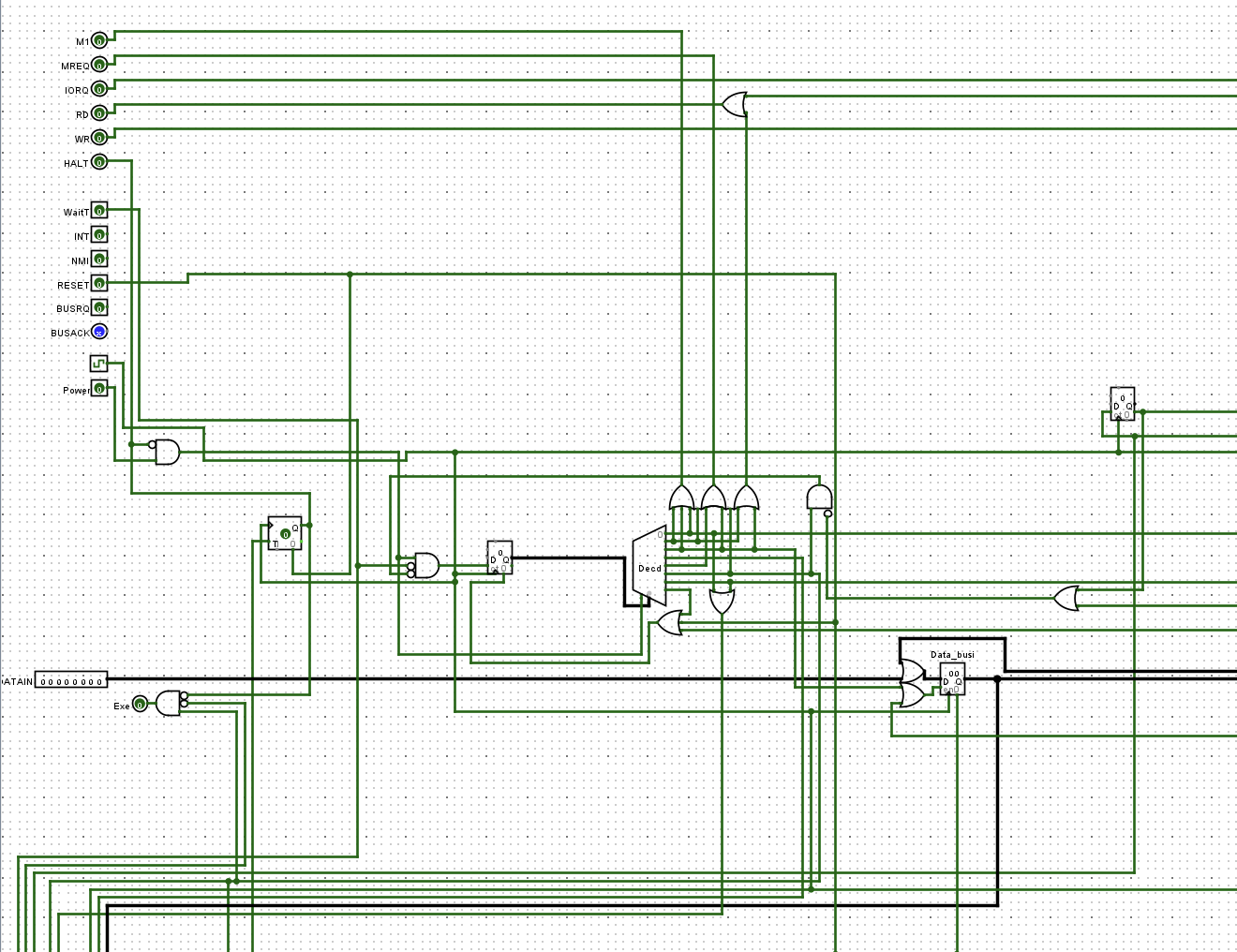
While **Data Out** after data has been processed inside the CPU gets directed to output either redirected to RAM for being stored or for being printed in a Output device .

A screenshot of a computer

Description automatically generated

# CPU

After combining all the parts, thought the project , including the Control logic and Instructions Decoding Systems the General Purpose Registers, Flag registers and all other parts here is the final CPU:  
  
  
  


And the parts of the CPU in more detail first for the Data IN and Control part :  
  


The Control Logic and Instruction decoding system:

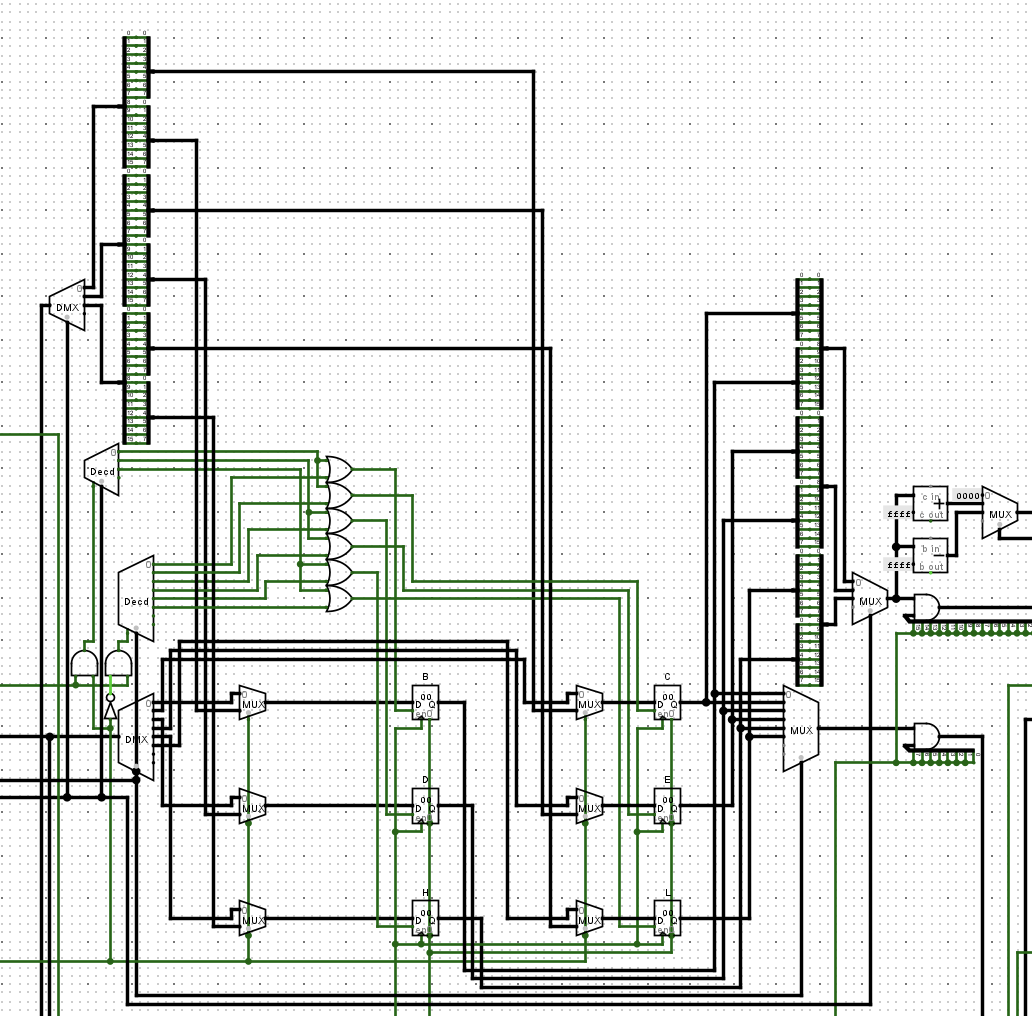
A green and grey rectangular object

Description automatically generated with medium confidence

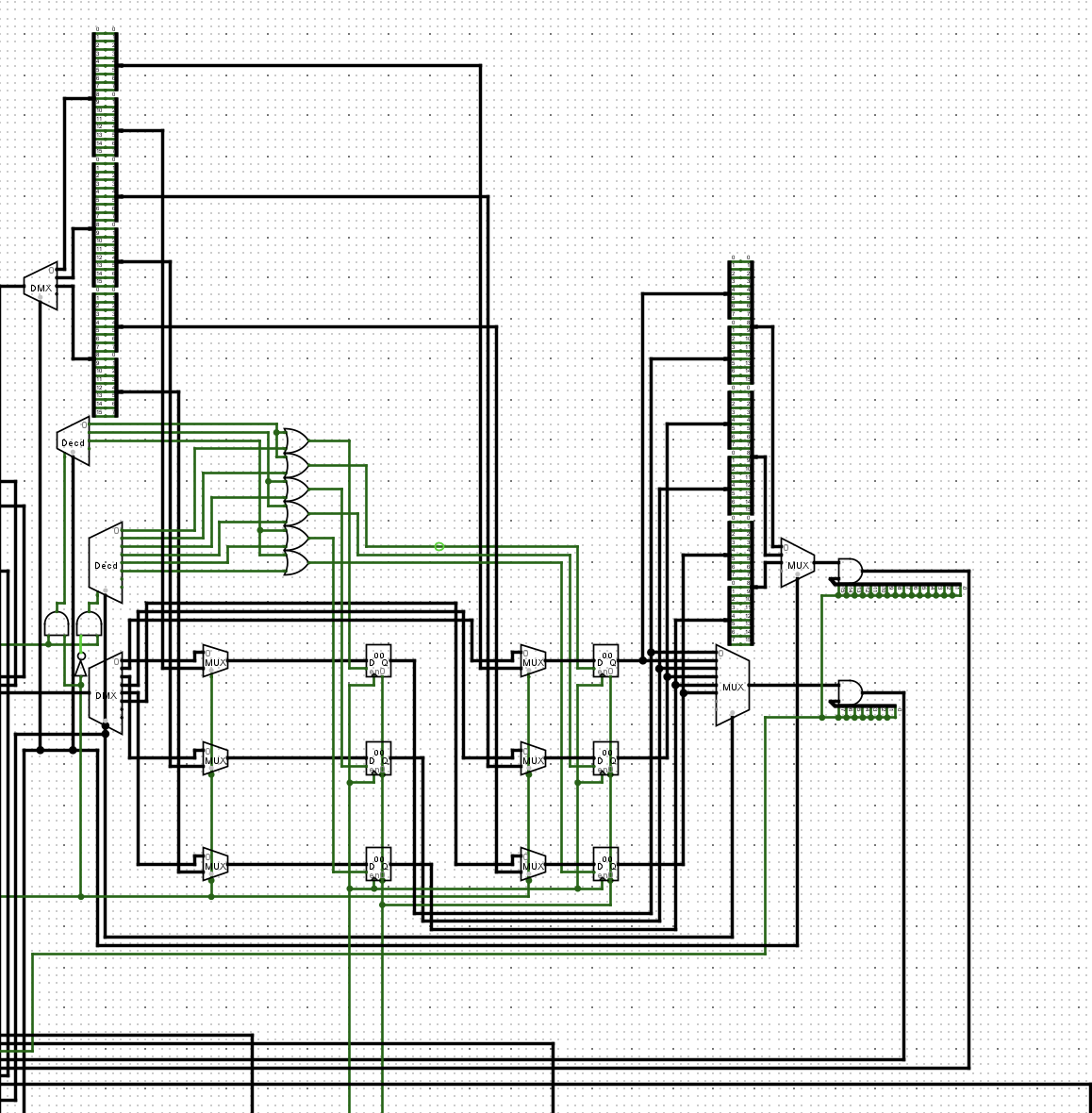
A diagram of a circuit

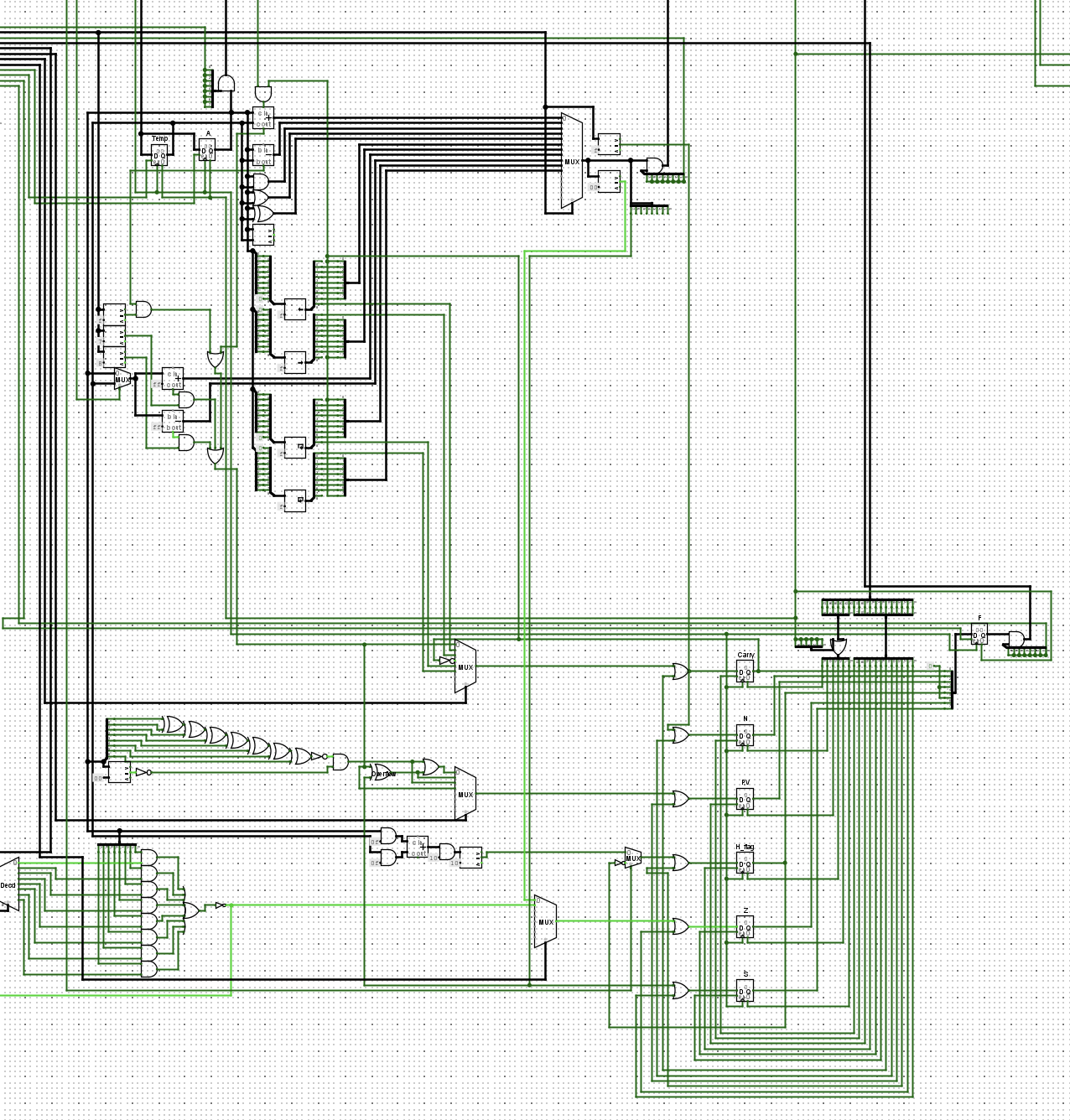
Description automatically generated

The General Purpose registers(GPR):

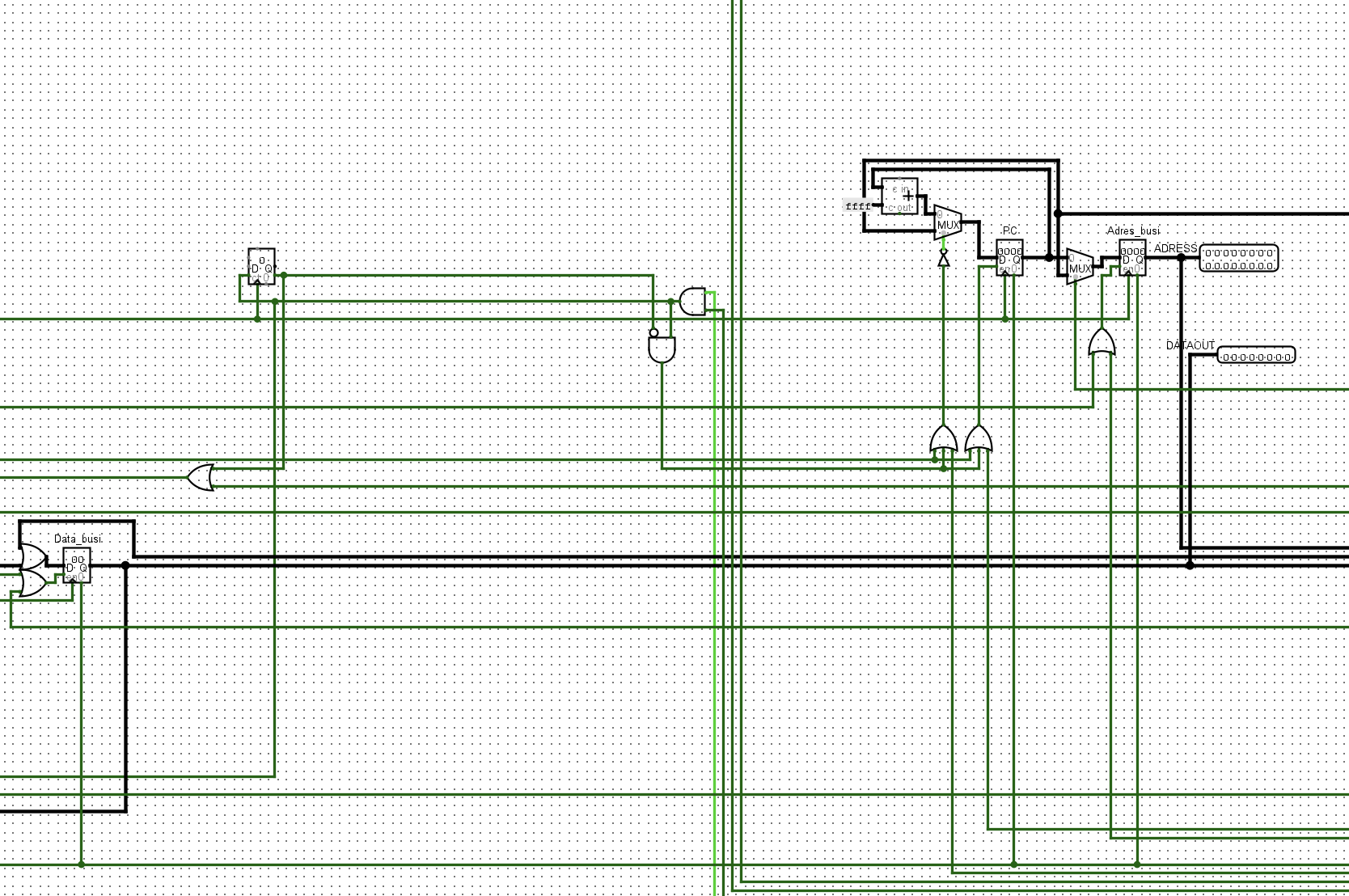


And the other registers similar to GPR:



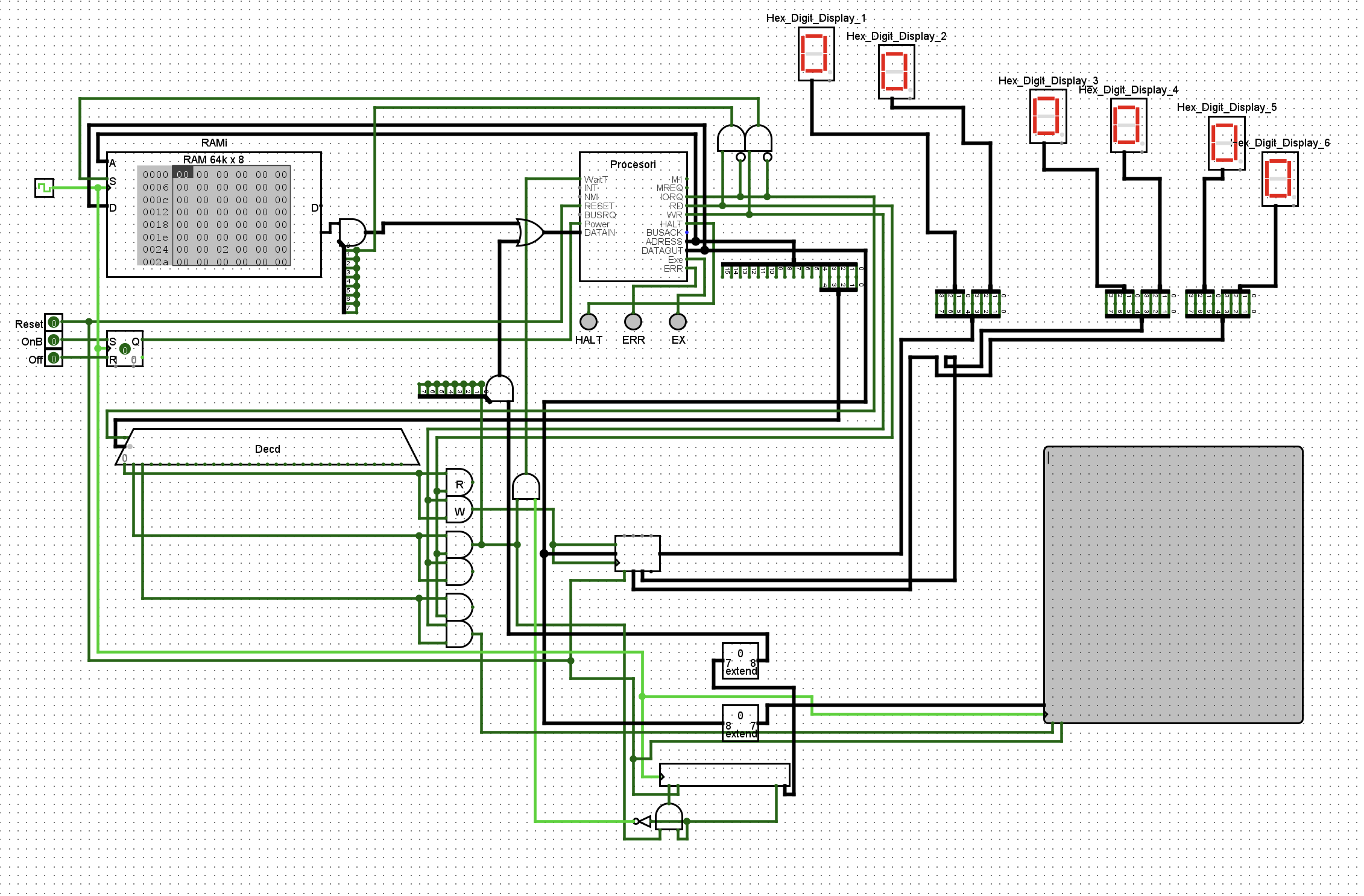
ALU:  
  


Address Bus, Data Bus and data out:



# Computer with the custom Processor and RAM

After successfully completing the processor, I continued and integrated also the RAM and other components including a Teletypewriter (TTY), 8 Digit Displays ,2 bit extenders , a decoder , 12 AND gates, 1 OR gate , 1 shift register:



# Instructions

After development of my custom CPU based on Z80 architecture , I was able to run some instructions to test it and here are the instructions I ran:  
  
The instruction “**add x,y”**  that adds the contents of register y to register x and stores the results in x.   
Also instructions like “**ld z,x**” and “**ld y,x**” for loading the data , transferring the data between the register where in the “**ld z,x**” loads the contents from x to z and the “**ld y,x**” the contents of x to y.  
Some other instructions I ran :  
ld y,\*  
inc x   
inc y

and inc z

# Conclusion

In this project I successfully designed and implemented a custom CPU based on Z80 architecture. In this project design I included parts like a custom control logic and instruction decoding unit for a expanded instruction handling. A 8-bit-general-purpouse register . Also along side with more specialized 16-bit address registers that includes Program Counter(PC) , Stack Pointer(SP) , IX , IY. Also a working ALU supporting addition , subtraction , shifting , rotation and comparisons and flags to track arithmetic outcomes. And in the end after designing and implementing the Z80 based CPU I integrated also with RAM and a basic I/O devices to form a complete “computer”

# References

**Z80 Documentation:**  
https://www.zilog.com/docs/z80/um0080.pdf

**Z80 instructions:**

<http://z80-heaven.wikidot.com/instructions-set:rr>

**Z80 Support page:**  
<http://www.z80.info>