****

**EE - 222**

**Microprocessor Systems**

**Assignment 1, 2 & 3**

|  |  |  |
| --- | --- | --- |
| **Name** | **CMS** | **Contribution** |
| Faizan Mansoor Aziz | 393817 | Design, Machine code |
| Raja Ibrahim Abdullah | 370239 | Design, Github |
| M Saqib | 373727 | Design, Github |
|  |  |  |

This assignment consists of design of a simple 4-bit Microprocessor on Proteus. Microprocessor consists of Program Counter, Instruction Decoder, ALU, 4 bit Registers, Flip Flop, Multiplexer, Decoder, 7 segment Display and AND,OR,NOT Gates.

**Block Diagram:**

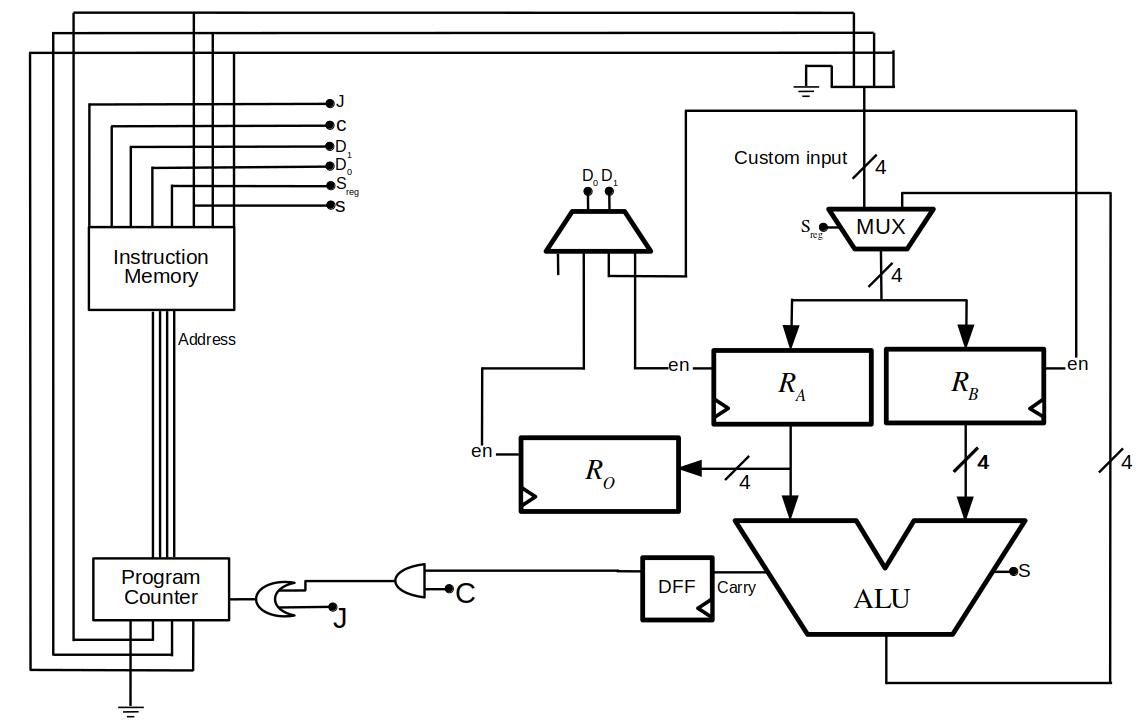


Figure 1:Block Diagram of 4 Bit Microprocessor

**Components:**

**Program Counter:** It is a 4-bit synchronous counter it counts from 0 to F after which it resets to 0 automatically. It has a “Load” input pin, when this pin is high, it inputs are custom inputs(for JMP).

**Instruction Decoder:** It generates addresses corresponding to the code that we have uploaded on it. It also enables/disables the components according to instruction.

**Quad 2-to-1 Mux:** It contains four 2 to 1 line multiplexer. It decides whether the inputs of circuit will be custom inputs or ALU outputs from previous instruction.

**2-to-1 line decoder:** It enables/disables respective register present in particular instruction.

**Registers:** There are three 4-bit registers for storing numbers and results of manipulation.

**ALU:** It contains Adder/Subtracter circuit. It has a control input S, when S=0 addition is performed and when S=1 subtraction will be performed.

**Seven Segment Display:** It is used to display contents of Ro in decimal form.

**Register Selection:**

There are 3 register in microprocessor design named Ro(Output Register), Ra and Rb. Contents of register can be read or write if they are enabled. There enable pin is connected to Decoder. Ro will be activated when inputs of decoder is 10 , Ra will be activated when input of decoder is 00 and Rb is activated when input of decoder is 01. On 11 no register with be activated so, nothing can be read or write in this case.

**Instruction Register:**



**Programming:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function** | **J** | **C** | **D1** | **D0** | **Sreg** | **S** | S | S |
|  | | | | | | **Custom Inputs** | | |
| *RA* = *RA* + *RB* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| *RB* = *RA* + *RB* | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| *RA* = *RA* − *RB* | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| *RB* = *RA* −*RB* | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| *RO* = *RA* | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| *RA* = *imm* | 0 | 0 | 0 | 0 | 1 | imm[2] | imm[1] | imm[0] |
| *RB* = *imm* | 0 | 0 | 0 | 1 | 1 | imm[2] | imm[1] | imm[0] |
| Jump to imm if carry out | 0 | 1 | 1 | 1 | 0 | imm[2] | imm[1] | imm[0] |
| Jump to imm | 1 | 0 | 1 | 1 | 0 | imm[2] | imm[1] | imm[0] |

**HEX File Contents:**

0A 1F 00 20 04 20 B2

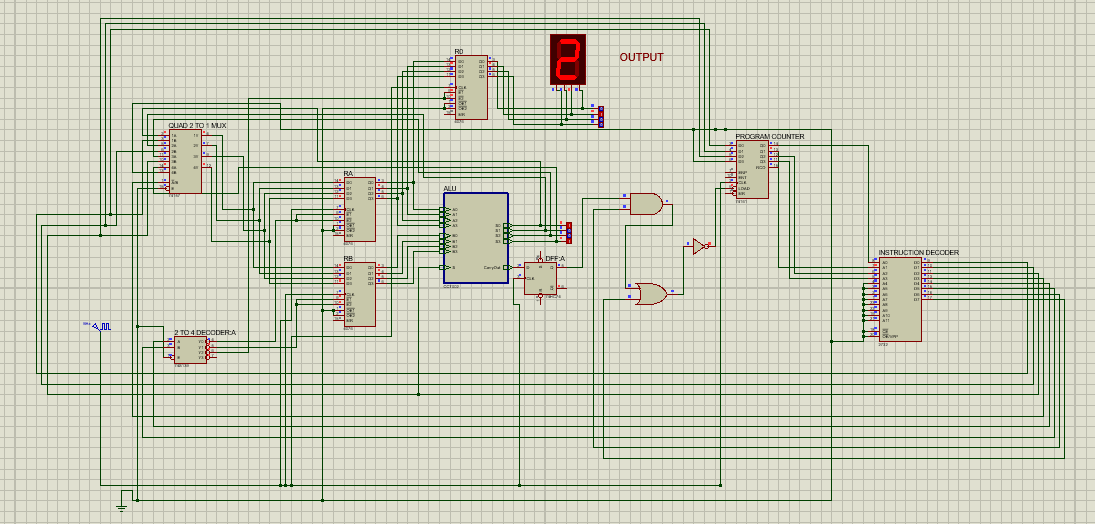
**Code Explanation:**

1. RA = 2 00 00 1 010 -> Load RA with 2 (LDI)
2. RB = 7 00 01 1 111 -> Load RB with 7 (LDI)
3. RA = RA + RB 00 00 0 000 -> RA = RA + RB (ADD)
4. RO = RA 00 10 0 000 -> RO = RA (MOVE)
5. RA = RA - RB 00 00 0 100 -> RA = RA – RB (SUB)
6. RO = RA 00 10 0 000 -> RO = RA (MOVE)
7. JMP to 1 10 11 0 010 -> JMP to 00 (JMP)

**Proteus:**

**A picture containing chart

Description automatically generated**

****