A picture containing circle

Description automatically generated**MARMARA UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING**

CSE 3015

Digital Logic Design

CPU Project

Name Student Number

Mustafa Tunahan BAŞ 150119055

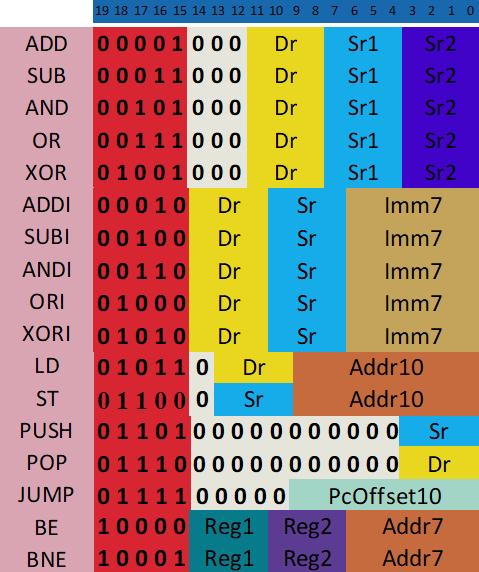
Mehmet Fatih ERDEM 150119054

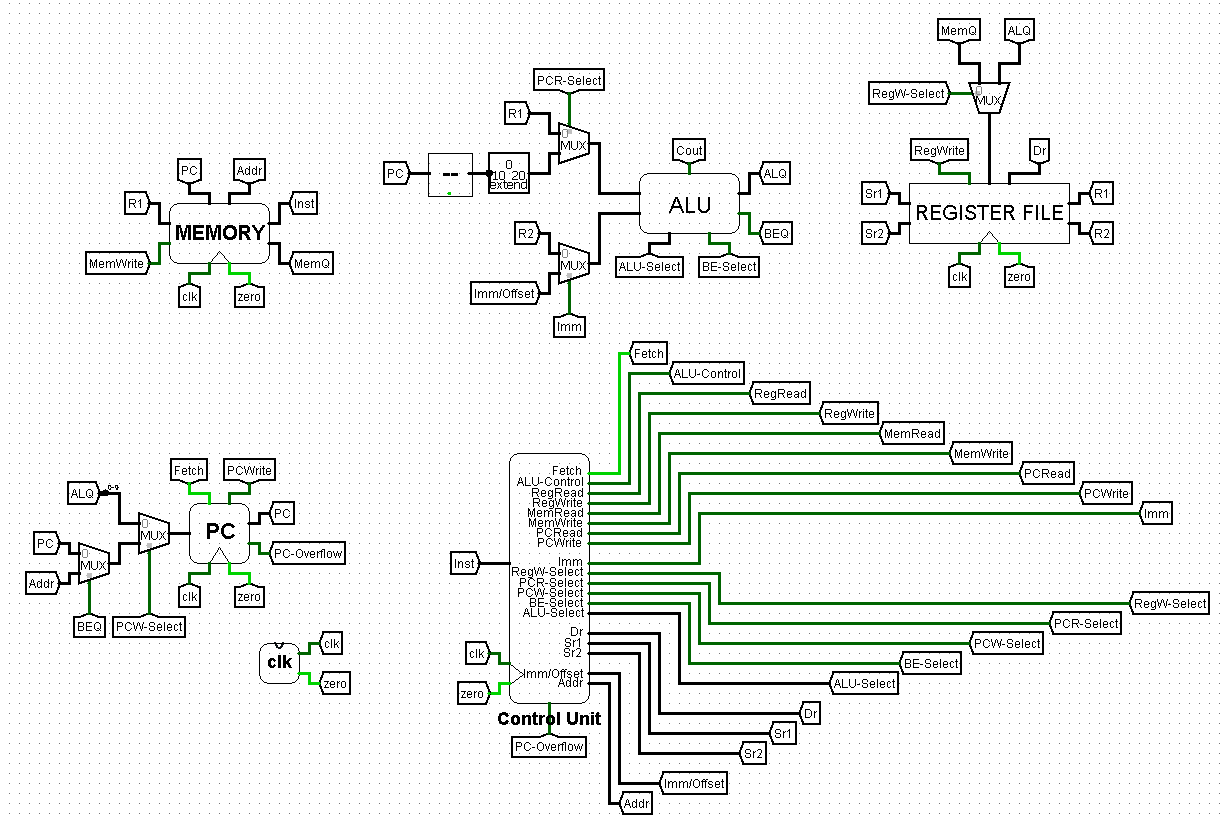
Talha SEZER 150119022

Mehmet AKGÜN 150119061

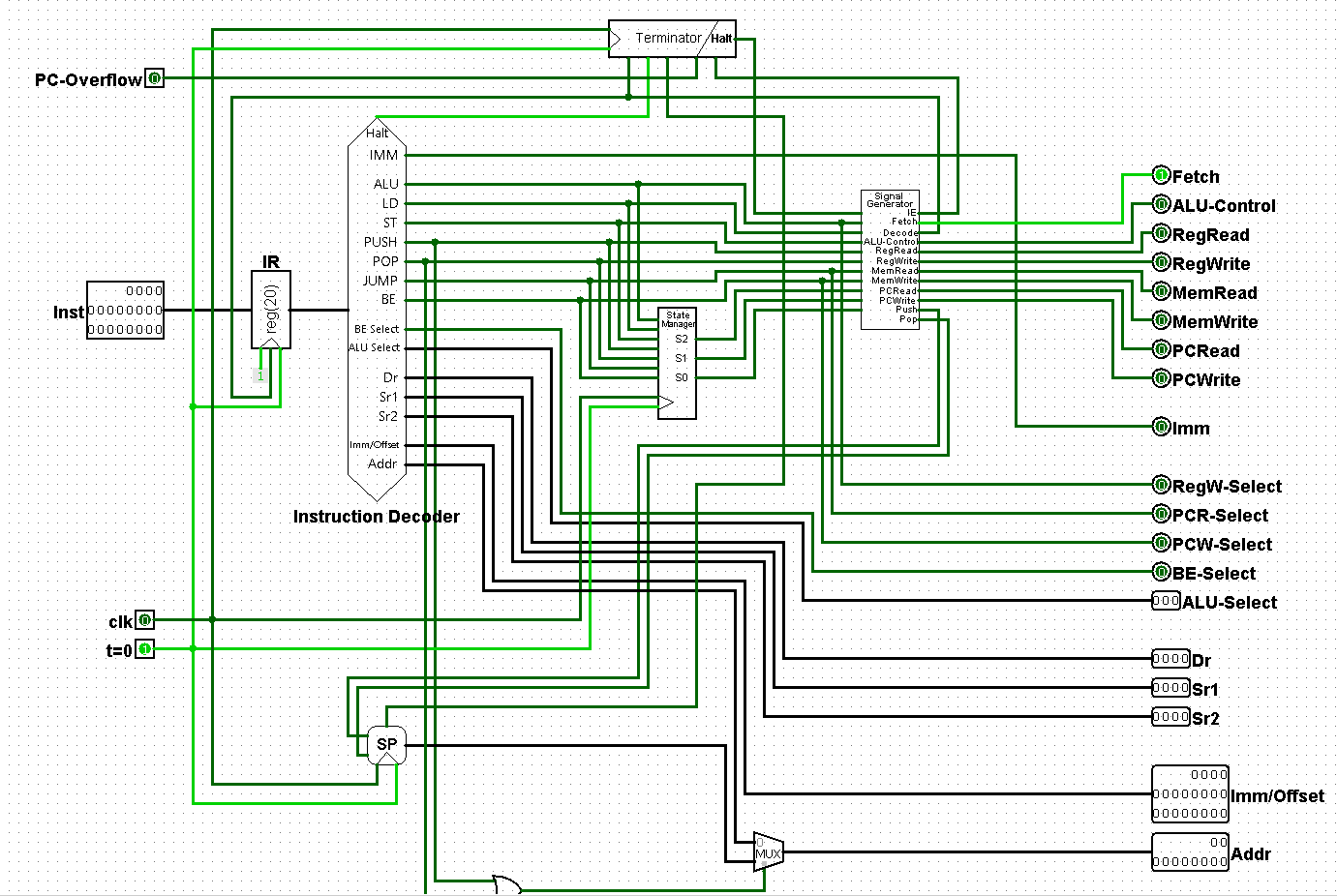
Submitted To : Betül BOZ

**Instruction Set Architecture (ISA)**

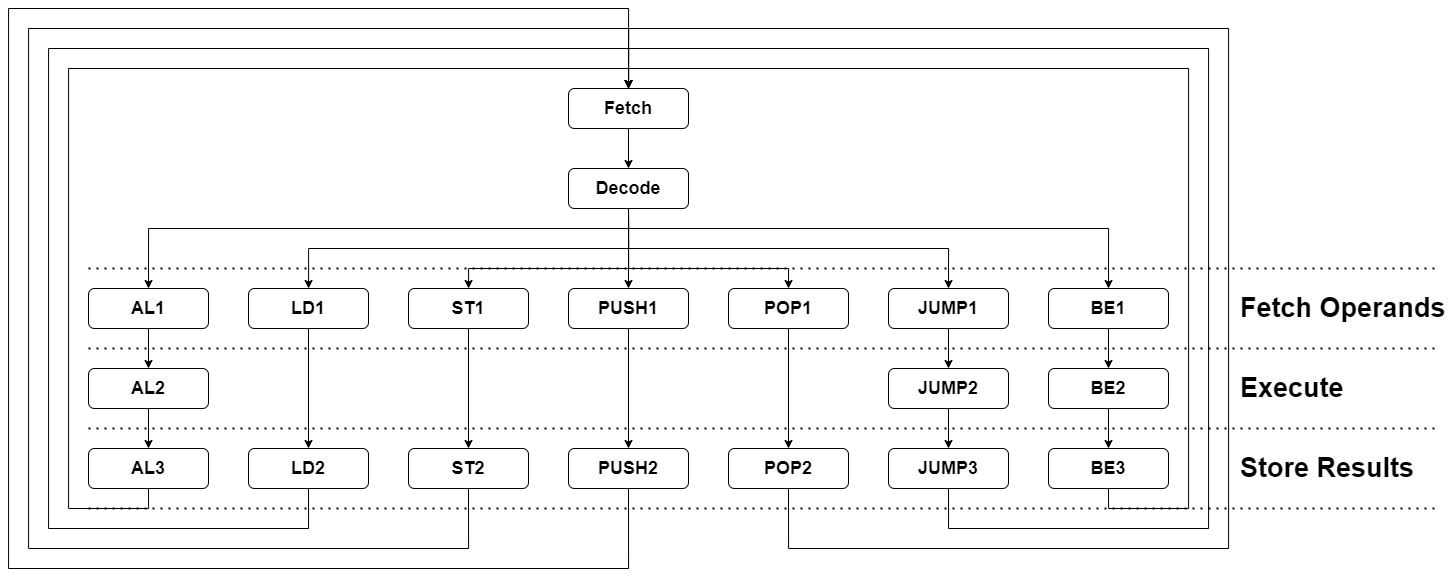
****

**CPU**

**Control Unit**

****

The control Unit consists of Instruction Decoder, State Manager, Signal Generator, Terminator and SP(Stack Pointer) subcircuits which help divide the complexity of the circuit. Control Unit has 3 inputs : clock, instruction, and PCOverflow. The Control Unit uses these inputs to output the required data (e.g. registers, addresses, select bits, signals) for the rest of the CPU. Here’s the state diagram for the Control Unit :



Each instruction takes 4 to 5 clock cycles, these cycles are divided into five phases : Fetch, Decode, Fetch Operands, Execute and Store Results. Since each instruction must be made of these five phases, the finite state machine in the Control Unit does not represent all of the states above, instead the FSM only represents the phases with 3 bits and the rest of the state information comes from the type of the instruction. For example the equivalent of the AL2 state in the state diagram would be AL & Execute(011) in the Control Unit. This means the state is actually represented by 10 bits instead of 5 bits that was necessary but it also means the Control Unit only need 3 flip flops instead of 5 to save its state.

**Instruction Decoder :** Instruction Decoder circuit takes the instruction from the IR(Instruction Register) and extracts all the needed information from it which are the type of the instruction (e.g. ADD,LD), source and destination registers, immediate values and addresses, and some select bits needed for the instruction’s flow. The Instruction Decoder consists solely of combinational logic.

**State Manager :** State Manager is a sequential circuit that takes as input the clock and the instruction type data generated in the Instruction Decoder and outputs the current state of the instruction while updating the state registers for the next state (This is also done in a subcircuit called Next State which just implements the combinational part of the circuit). Note that the state registers inside the State Manager do not represent the current state completely, they only represent the current phase of the instruction. Instruction type and state bits together represent the whole state.

**Signal Generator :** Signal Generator takes the current state data (state bits and instruction type combined) from Instruction Decoder and State Manager and Terminator (Halted state which prevents the Signal Generator from generating further signals) and outputs the signals necessary to execute the instruction, these signals are Fetch, Decode, ALU-Control , RegRead, RegWrite, MemRead, MemWrite, PCRead, PCWrite, Push, Pop, and InstructionEnd. The Signal Generator consists solely of combinational logic.

**Stack Pointer :** Stack Pointer takes the clock and Push and Pop signals generated by the Signal Generator and outputs the current address of the stack while also updating the address. Stack initializes at address 1023 and grows towards 0. Stack Pointer also outputs a StackOverflow signal when the address in the register overflows.

**Terminator :** Terminator takes the clock, Decode, HaltInstruction, StackOverflow, InstructionEnd, PCOverflow and determines whether the instruction execution must be stopped in which case it raises the Halted signal and saves it to an internal register which can not be overriden again so that the CPU will stay halted. The Terminator is a sequential circuit which uses a combination of rising edge and falling edge flip flops to save its state.

**Arithmetic Logic Unit (ALU)**

The ALU consists of 20 bit Adder, Subtractor, Bitwise And, Bitwise Or, Bitwise Xor and Equality Comparator subcircuits. It takes the 3 bit Op-select, two 20 bit inputs I0 and I1 from which the selected operation’s result will be outputed, and the Be-select which decides whether the Equality Comparator should compare for equality or inequality. The outputs are the 20 bit result of the operation selected, cout (if one of Adder or Subtractor is selected) and e (the 1 bit output of Equality Comparator).

**Register File**

The Register file contains 16 registers of 20 bits. It has one write and two read ports (20 bits), along with a 4 bit W-select, R1-select, R2-select ports. It also takes the clock and W-enabled inputs. Clock is necessary for writing to the registers whereas reading doesn’t require the clock.

**Program Counter (PC)**

The Program Counter takes a 10 bit address, the clock, Fetch, and PCWrite as input and outputs the address of the current instruction and PCOverflow signal. Whenever Fetch signal is generated, the PC will increment its value by 1 whereas a PCWrite signal will cause the PC to take the value of the address inputted to it. If the value of the PC overflows during the Fetch signal, the PCOverflow signal will be generated and sent to the Control Unit to stop the execution.

**Memory**

The Memory circuit contains two RAMs inside it both of which have separate load and store ports. One of the RAMs is used as data memory only whereas the other one is instruction memory only. The memory circuit takes two 10 bit addresses (one for each RAM) and a 20 bit Data and WR-select for the data memory. It outputs two 20 bit data (one from each RAM).

**clk**

The clk circuit contains the clock in it and also another circuit called Tick Once At Zero that outputs 1 until the first clock signal, all the circuits that contain sequential logic use this signal to reset the flip flops before the first clock signal arrives.