# ECE 486/586

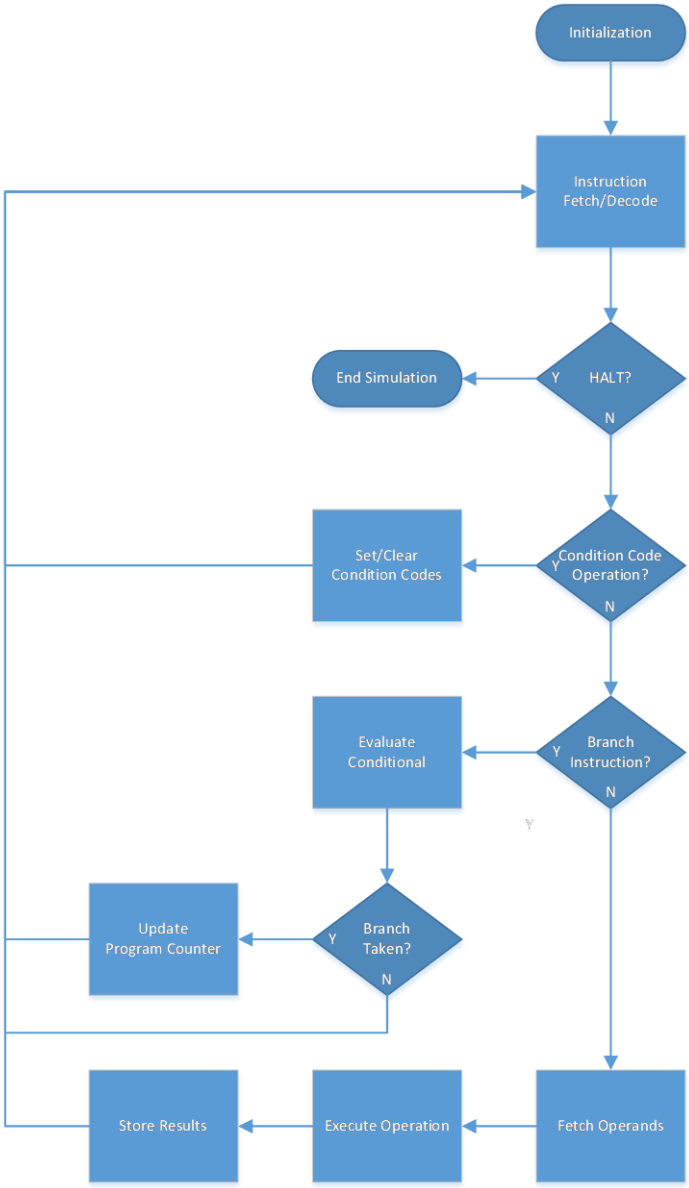
# PDP-11/20 Simulator

**Jordan Fluth, Brett Dunscomb, Scott Lawson and Robert Gaskell**

## Team Member Roles

Jordan Fluth: Instruction fetch and decode, RAM data writes.
Brett Dunscomb: Instruction execution, addressing modes and data fetch.
Scott Lawson: File IO, memory initialization and simulator testing.
Rob Gaskell: Source code file structure, main program loop organization, addressing modes, data fetch and instruction execution.

## Simulator Design




The PDP-11/20 Simulator imitates the behavior of the PDP-11/20 instruction set architecture, excluding instructions which operate on byte-length operands and those related to the interrupt mechanism.

The simulator begins with a function that reads an ASCII-converted object file and stores the included instructions and data to a memory array. If initialization fails, or the ASCII file is not found, the simulator exits.

Instructions are then fetched and decoded, and the Program Counter is incremented based on the length of the instruction. If a HALT instruction is fetched, the simulator ends.

Fetched instructions are categorized as Condition Code, Conditional Branch, Subroutine Jump, Single-operand, Double-operand, or Special Double-operand operations. For Condition Code operations, the appropriate Condition Codes are set or cleared, and the instruction at the address in the Program Counter is then fetched. For Conditional Branch instructions, the conditional case is evaluated, and if true, the Program Counter is set to the Branch Target address. Otherwise, the Program Counter is

unchanged. The instruction at the address in the Program Counter is then fetched. For all other operations, the operands are fetched from memory, the operation is executed on the operands, and any results are written back to the registers or memory, based on the addressing mode applied to the operands. The instruction at the address in the Program Counter is then fetched, as with other operations, repeating the program loop.

## Using the Simulator

The simulator reads an ASCII-converted object file, containing code and data for simulated execution. The simulator is run from a command line, with options configured via the following arguments:

| | |
|---|---|
| -f [filename]: | specify input filename |
| -c [PC]: | specify initial Program Counter value |
| -o [path/filename]: | specify output trace file directory/filename |
| -r [path/filename]: | specify register dump file directory/filename |
| -d: | enable print of GPR contents to stdout |

## Source Code

The source code for the simulator is attached, and can be found here:
https://github.com/JustRob83/pdp11/tree/master/src

## Test Plan