EDE

ELB816 Development Environment

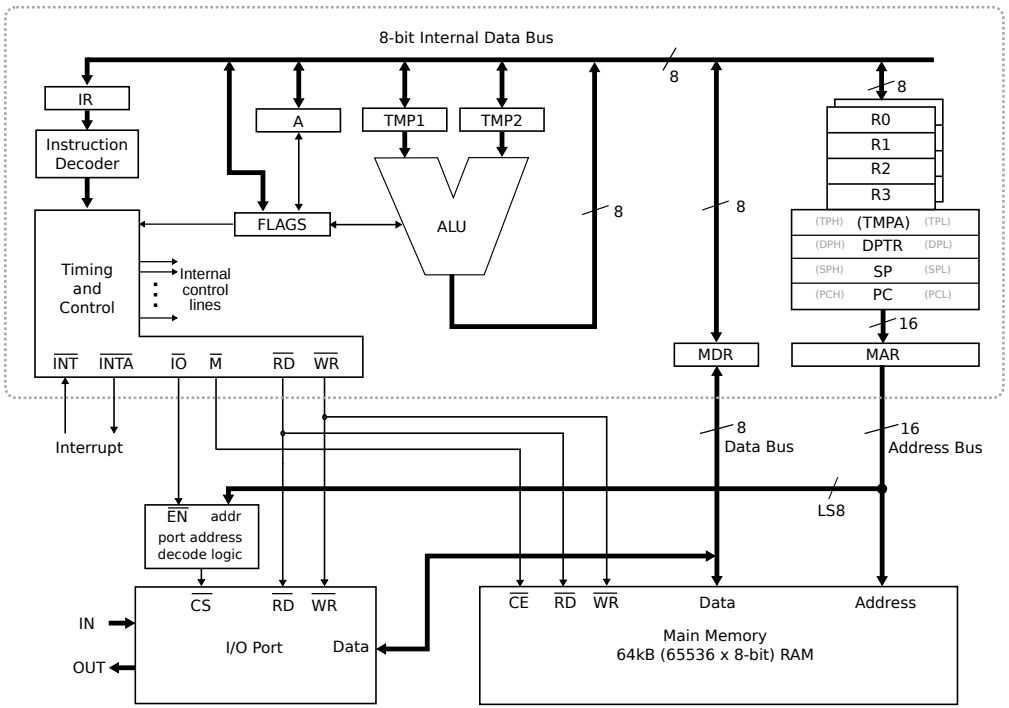
James Bowden (110104485)

Electrical and Electronic Engineering

April 25, 2014

Outline

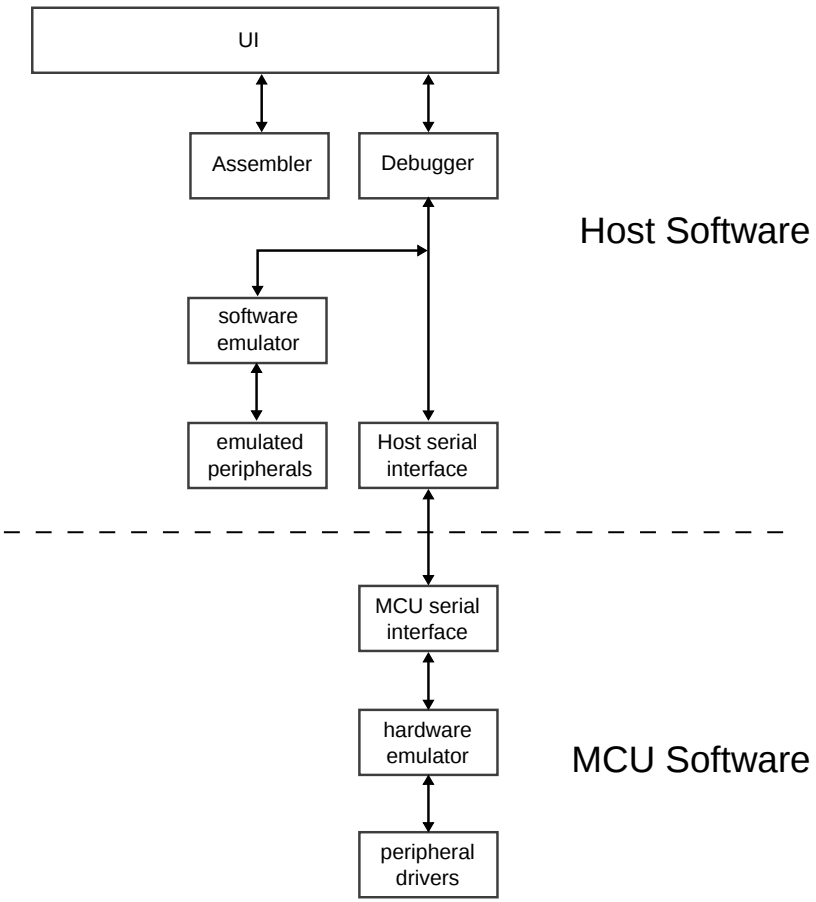
What is ELB816?



from the original ELB816 Specification [1]

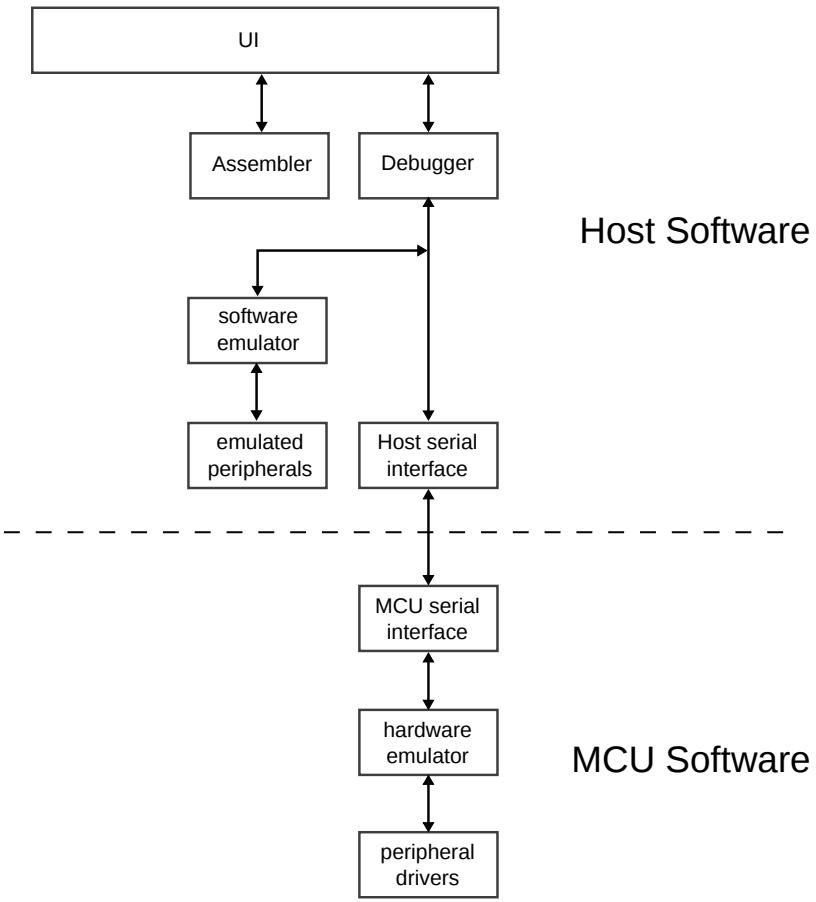
- "A simple to understand 8-bit microprocessor system to help people learn about microprocessor electronics."

What is EDE?



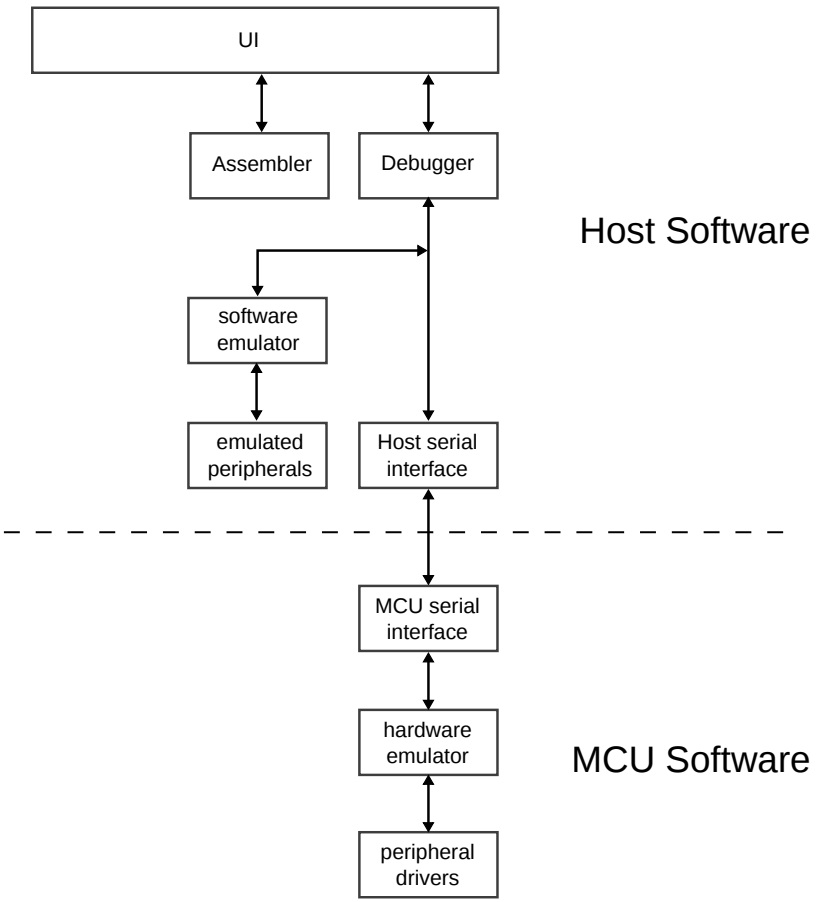
- A software implementation of the ELB816 microprocessor along with a development tool-chain
- An Assembler
- An Emulator
- A Debugger

What is EDE?



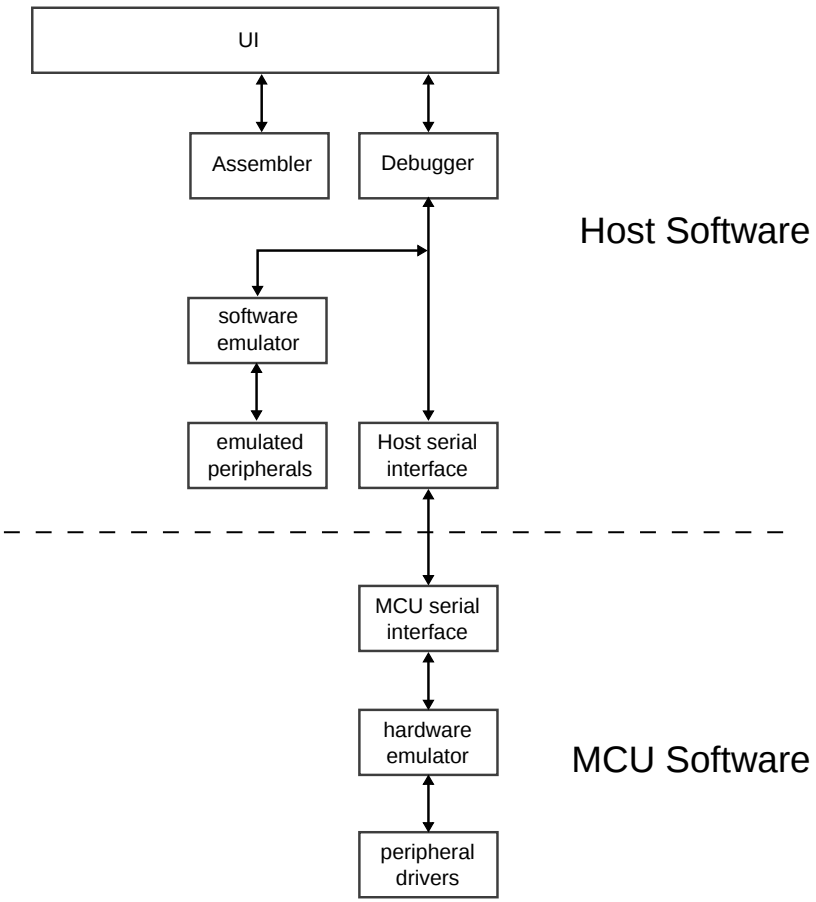
- A software implementation of the ELB816 microprocessor along with a development tool-chain
- An Assembler
- An Emulator
- A Debugger

What is EDE?



- A software implementation of the ELB816 microprocessor along with a development tool-chain
- An Assembler
- An Emulator
- A Debugger

What is EDE?

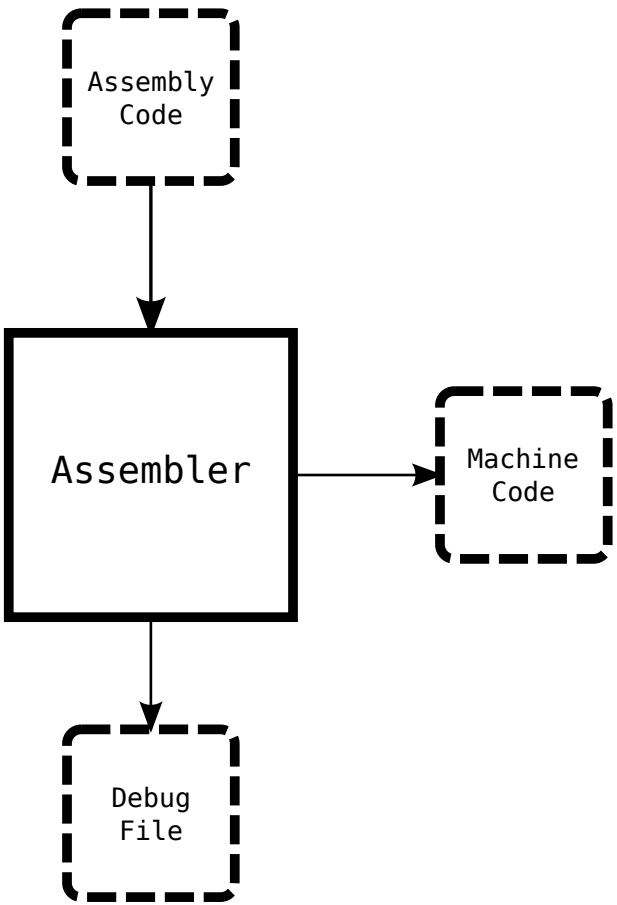


- A software implementation of the ELB816 microprocessor along with a development tool-chain
- An Assembler
- An Emulator
- A Debugger

What is the point?

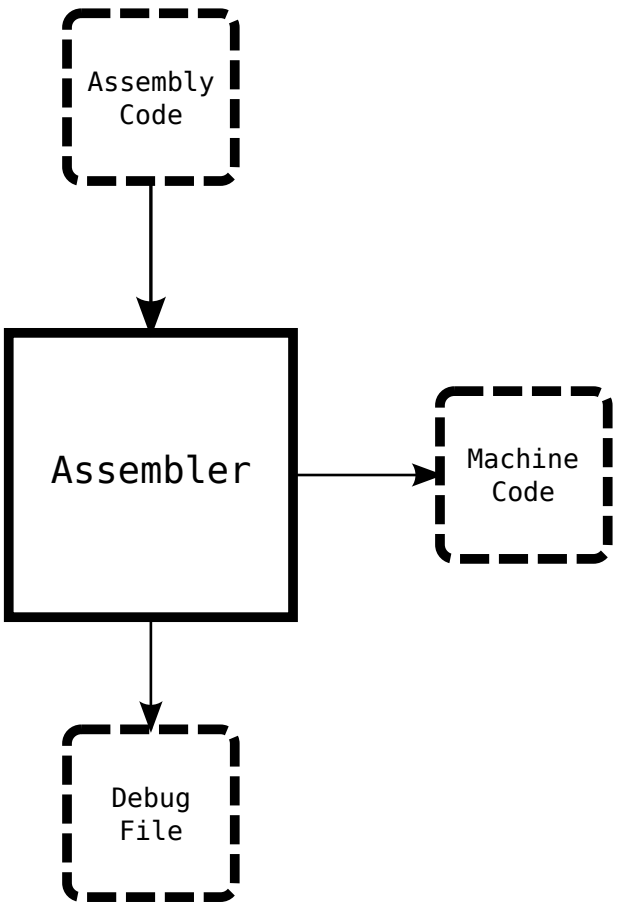
- To provide an environment for teaching and learning about micro-processor programming
- To provide an environment free of the the intricacies of real world micro-processors

The Assembler



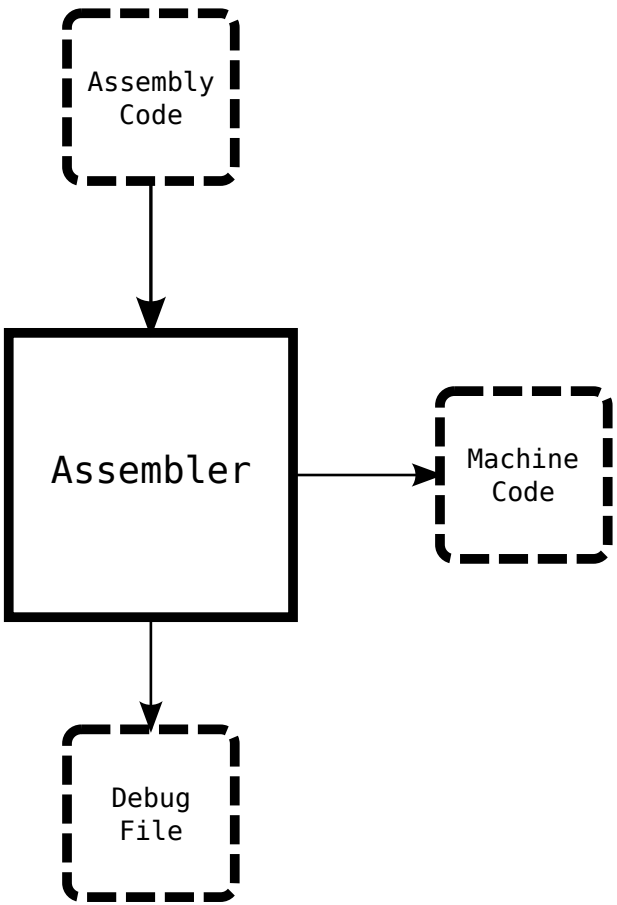
- Written in Python using only standard libraries
- Two pass assembler
- Produces raw binary executables and “debug files”

The Assembler



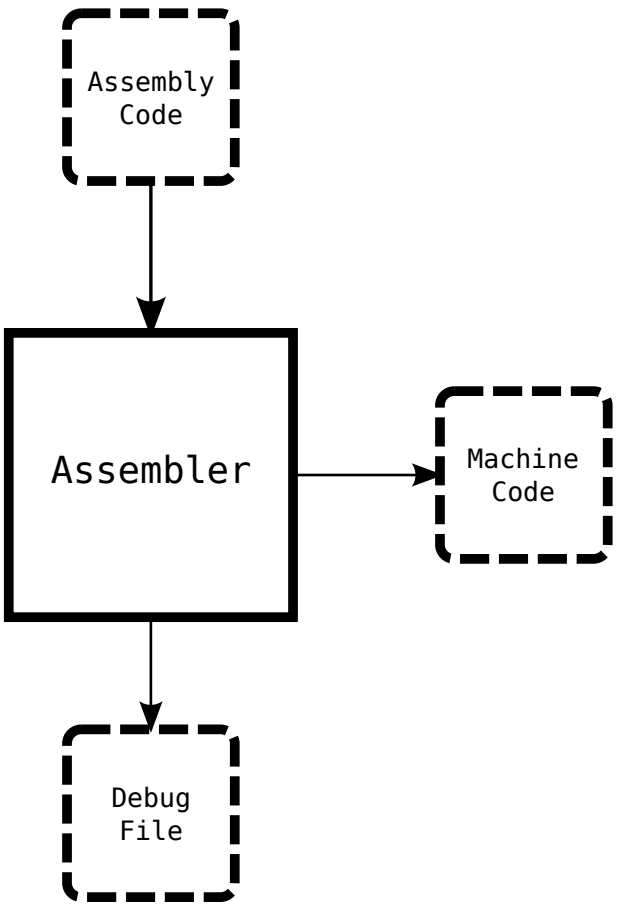
- Written in Python using only standard libraries
- Two pass assembler
- Produces raw binary executables and "debug files"

The Assembler



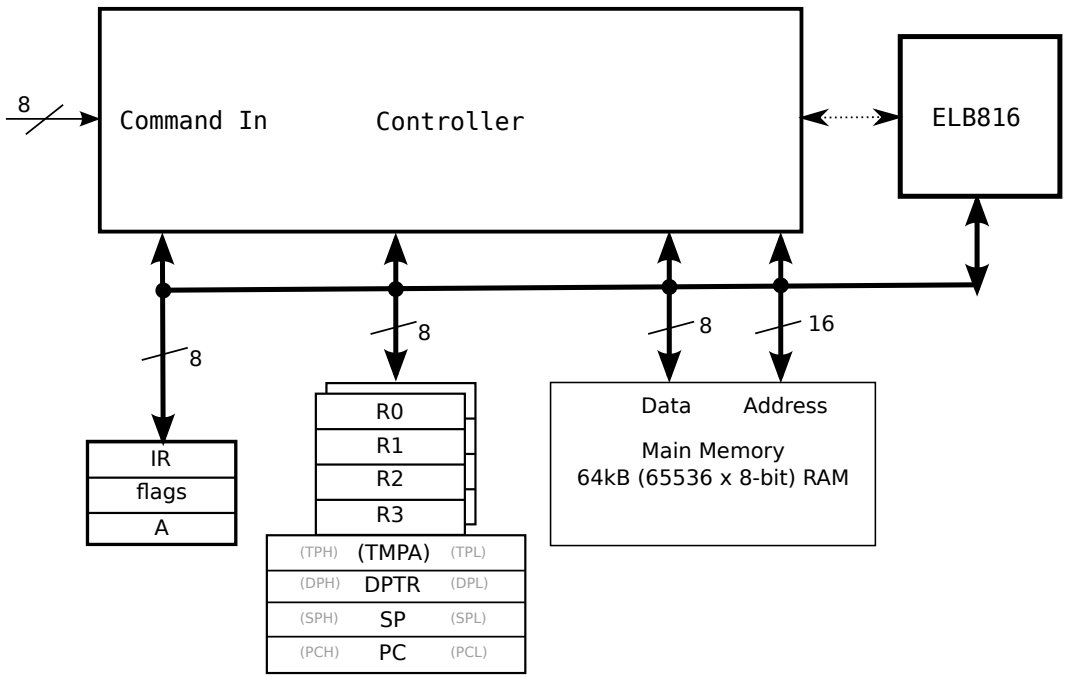
- Written in Python using only standard libraries
- Two pass assembler
- Produces raw binary executables and "debug files"

The Assembler



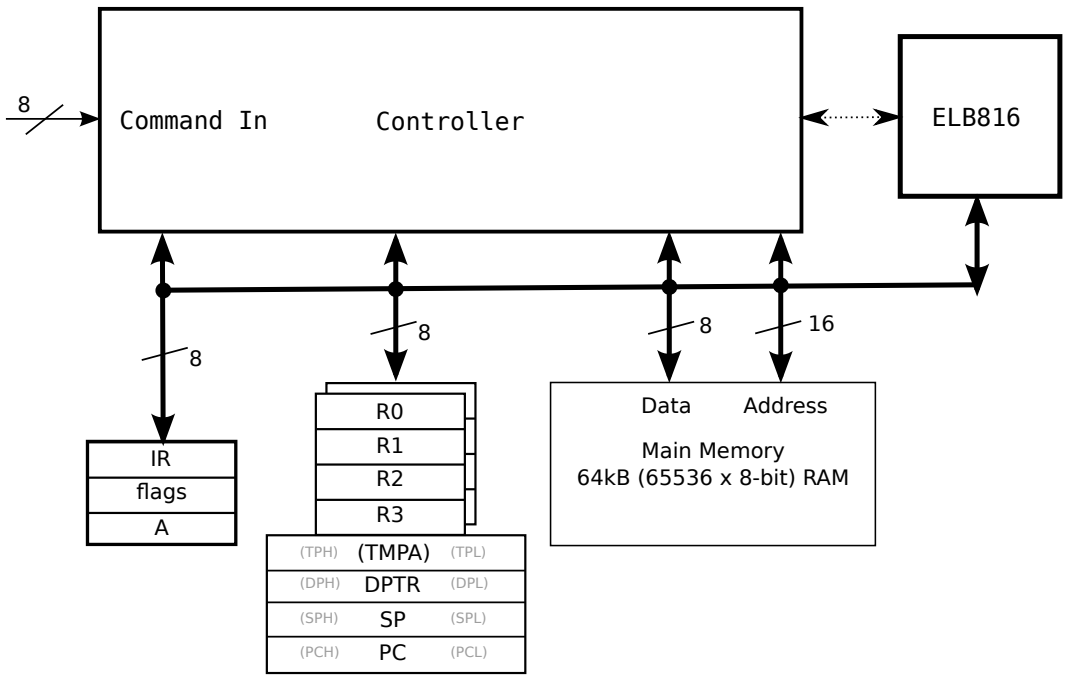
- Written in Python using only standard libraries
- Two pass assembler
- Produces raw binary executables and “debug files”

The Emulator



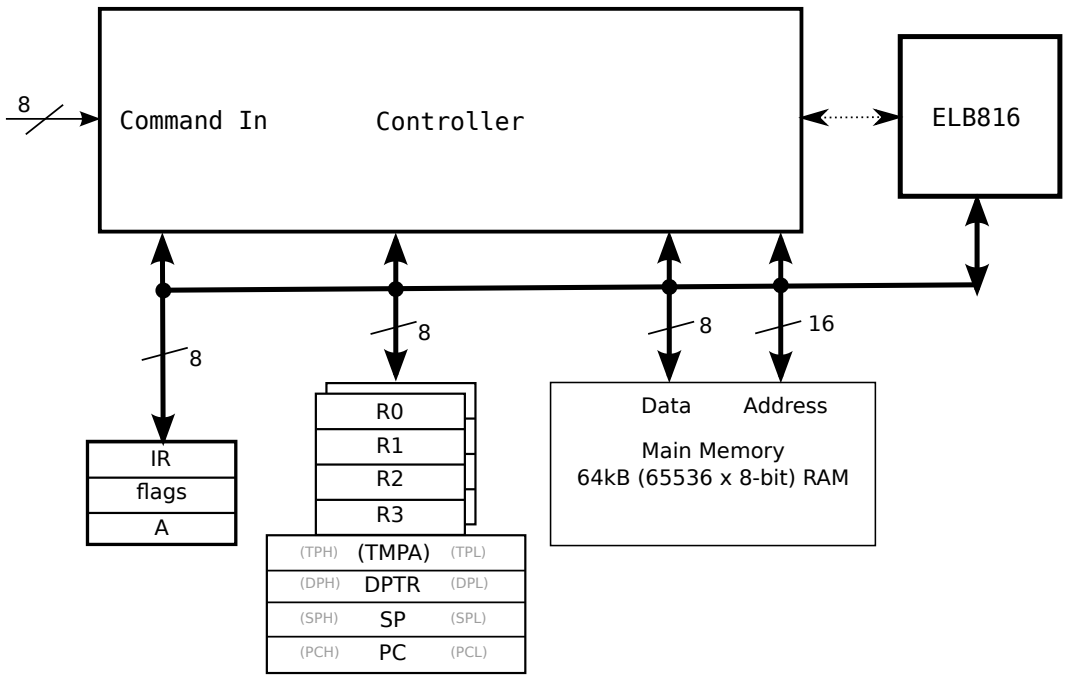
- Written in pure C89 using only standard libraries
- Separate builds for Linux and Intel MCS-51

The Emulator



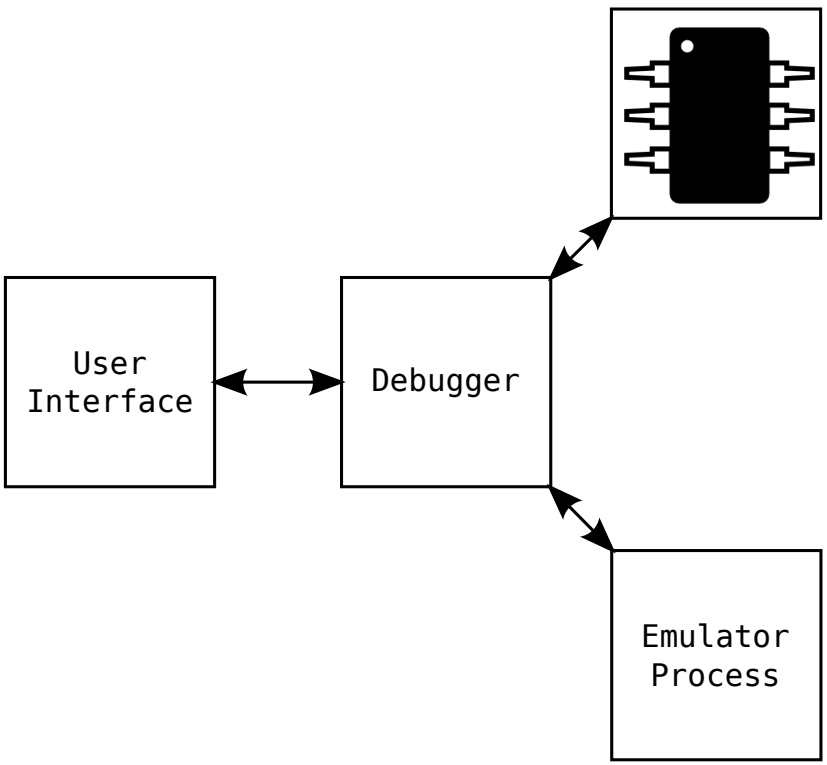
- Written in pure C89 using only standard libraries
- Separate builds for Linux and Intel MCS-51

The Emulator



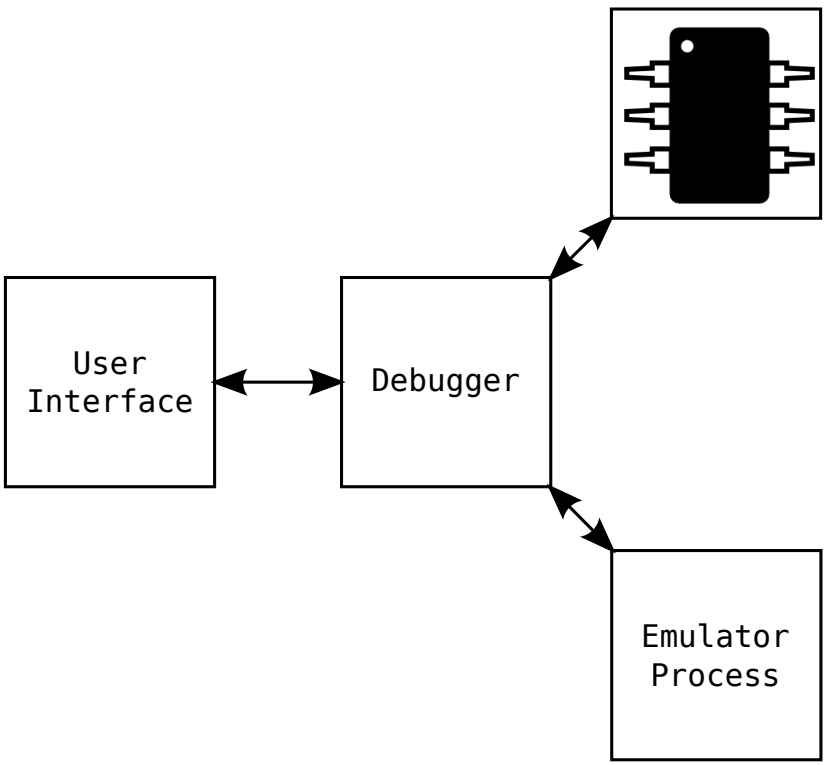
- Written in pure C89 using only standard libraries
- Separate builds for Linux and Intel MCS-51

The Debugger



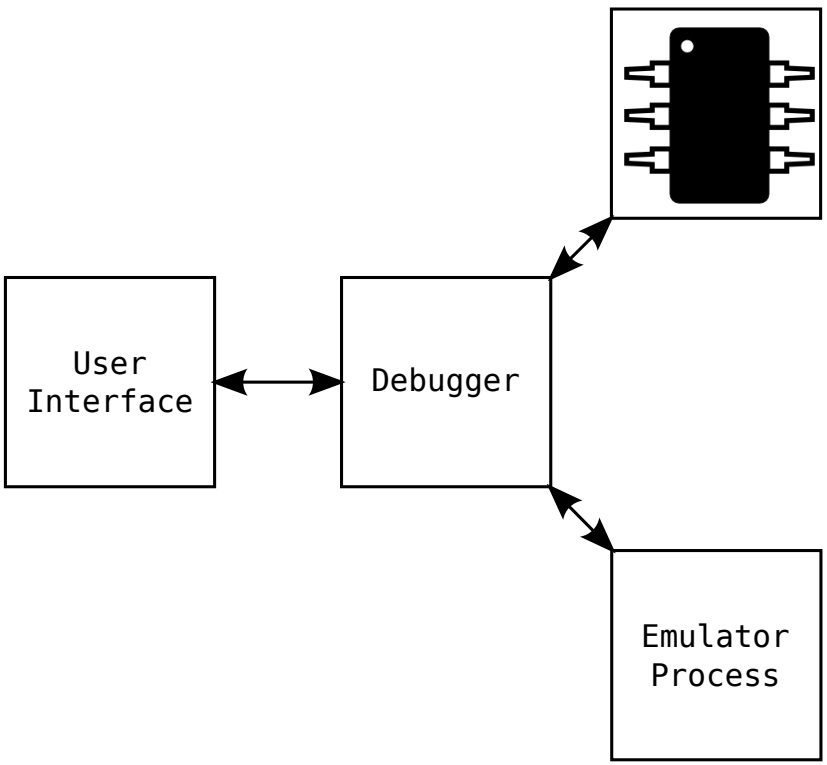
- Implemented as a Python class
- Communicates with the control function on a running instance of them emulator
- Text based command line interface built using this class

The Debugger



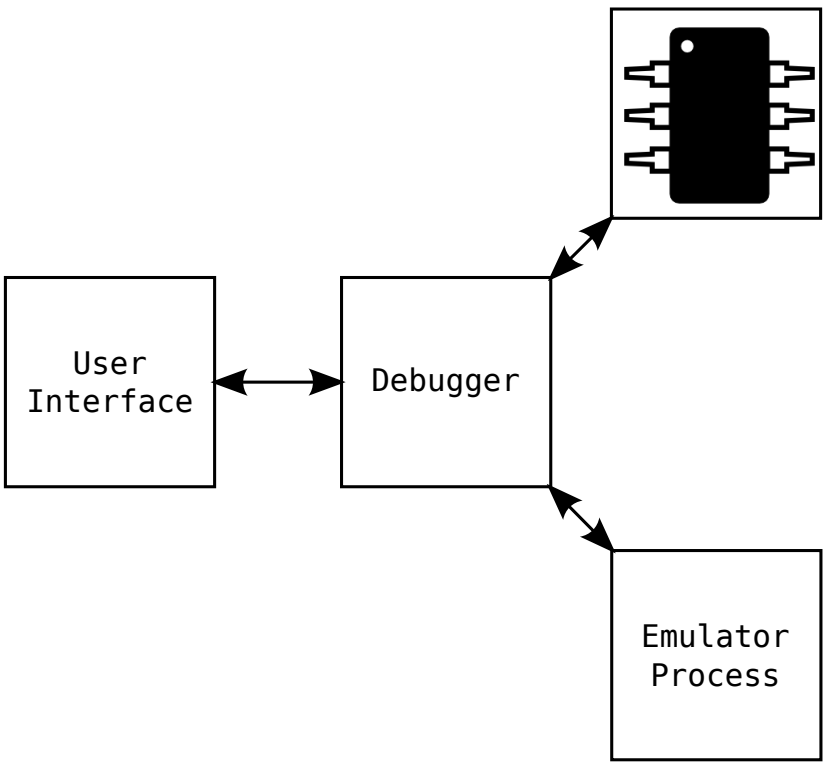
- Implemented as a Python class
- Communicates with the control function on a running instance of them emulator
- Text based command line interface built using this class

The Debugger



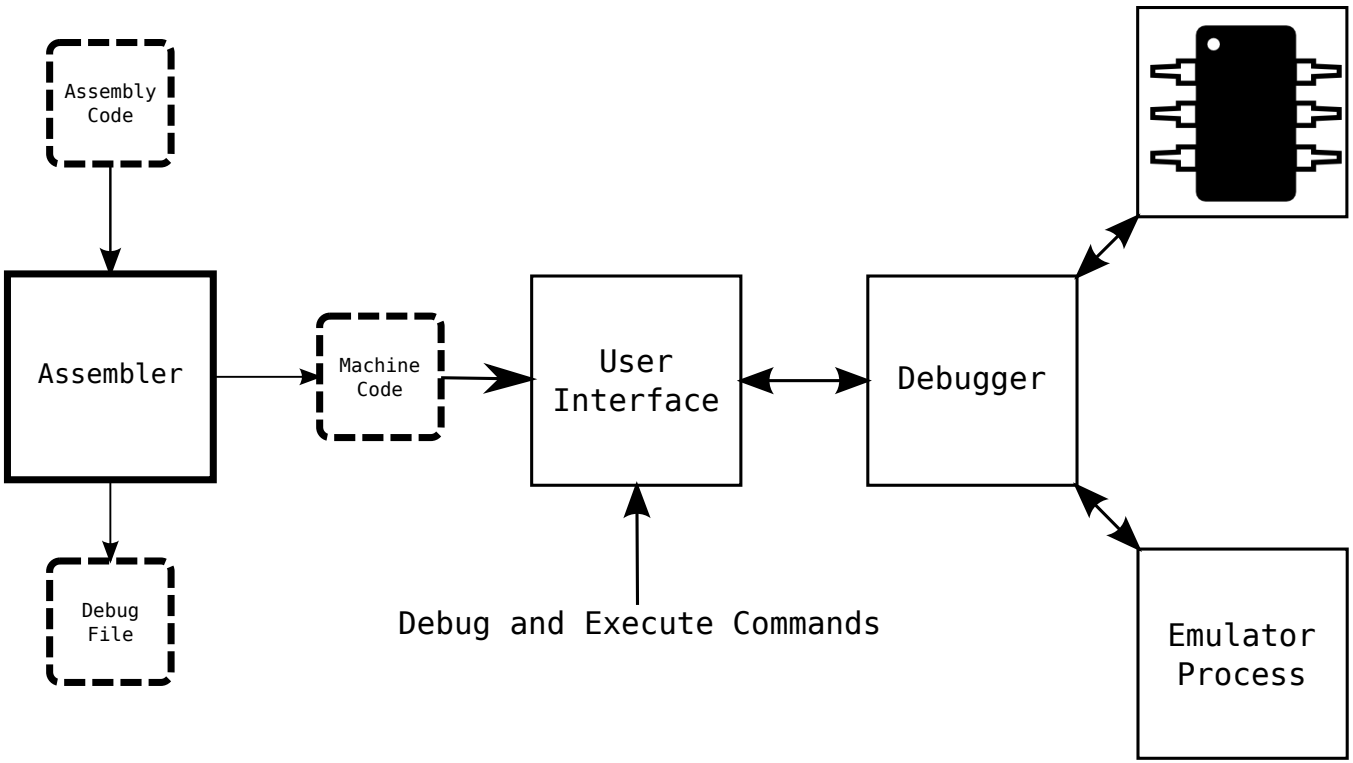
- Implemented as a Python class
- Communicates with the control function on a running instance of them emulator
- Text based command line interface built using this class

The Debugger

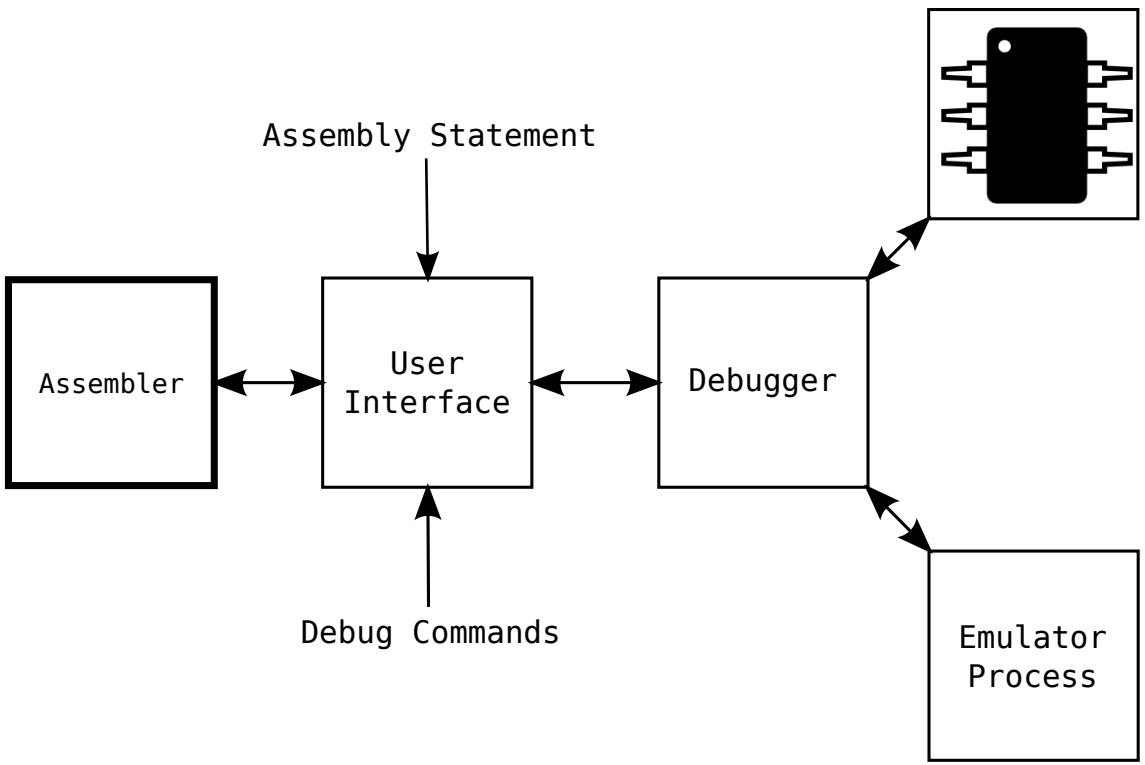


- Implemented as a Python class
- Communicates with the control function on a running instance of them emulator
- Text based command line interface built using this class

Assemble/Upload/Execute/Debug

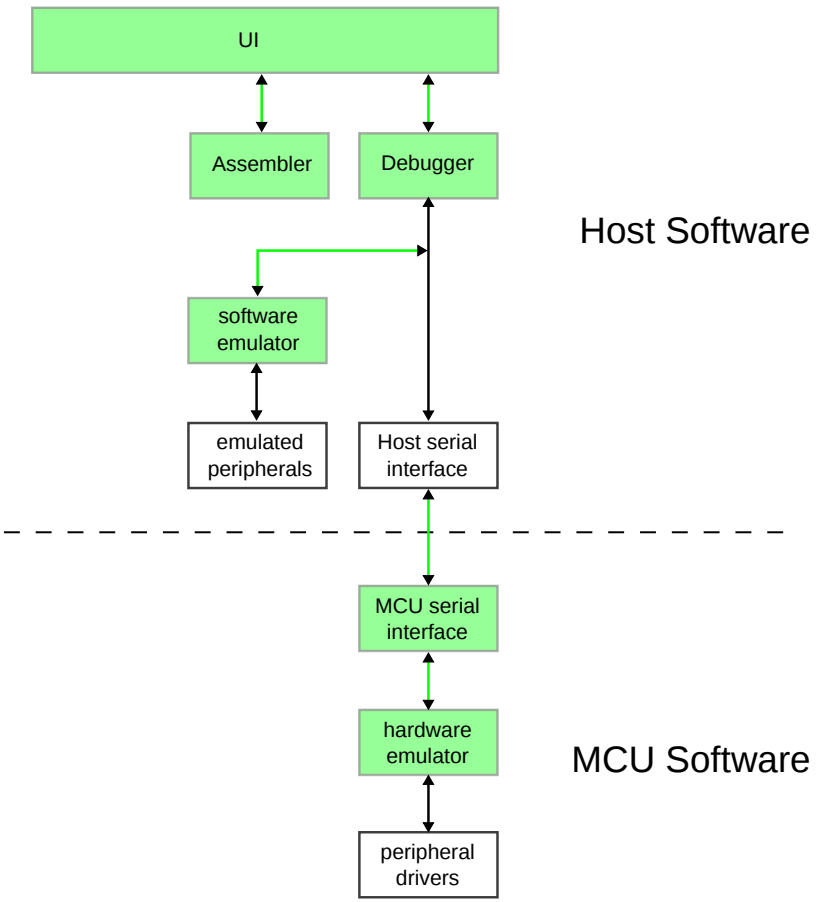


Interpreter Mode



Demonstration

Summary



- Developed a light weight portable software implementation of the ELB816 micro-processor core
- Developed a usable tool chain for writing and debugging code for the ELB816
- More to be done but code base is well commented and easily maintainable

References

1. ELB816 Home Page: http://code.google.com/p/elb816/