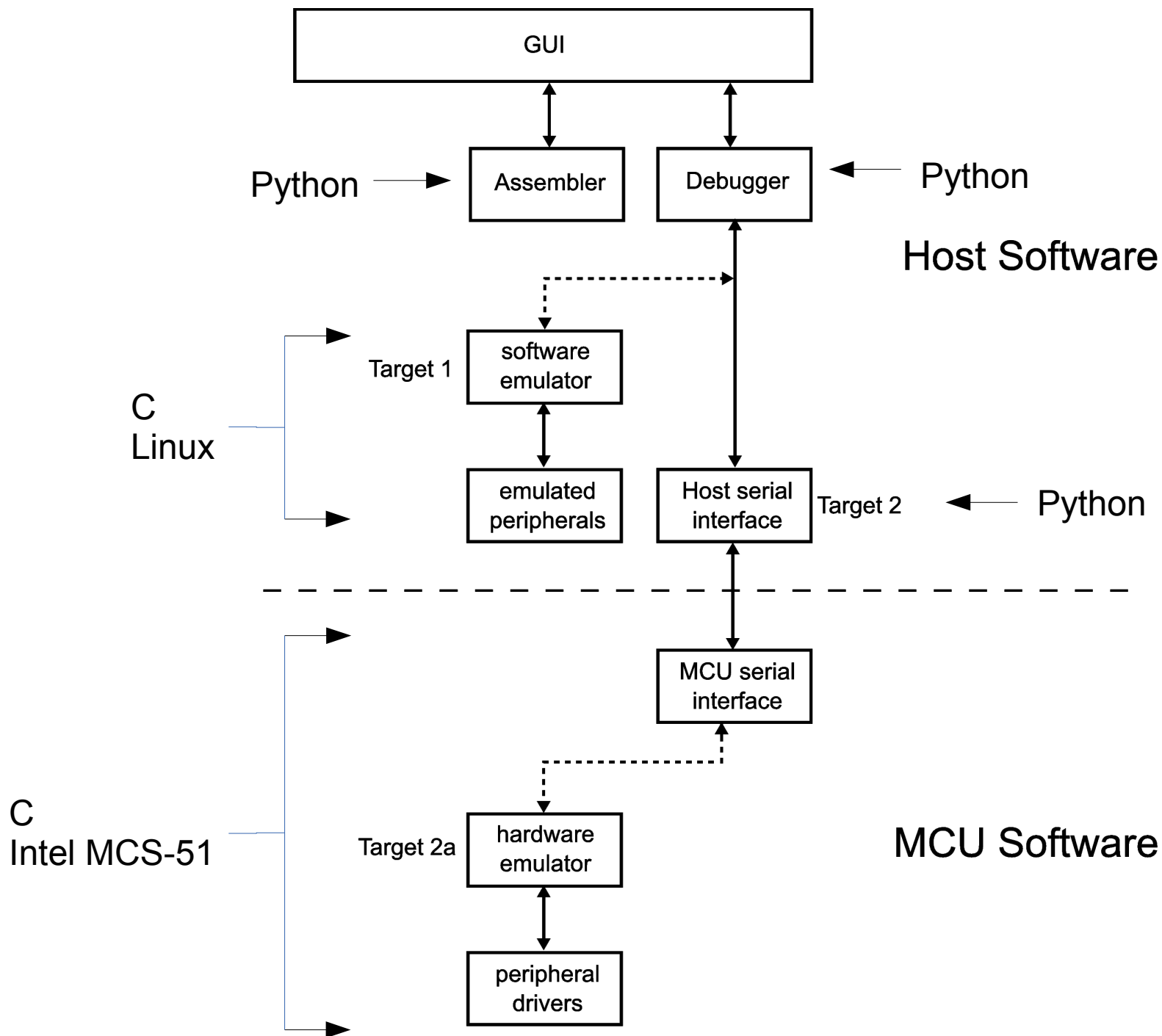


# ELB816 Development Environment

The ELB816 architecture is designed to be “a simple to understand 8-bit microprocessor system to help people learn about microprocessor electronics.”

EDE is a 'virtual' development environment consisting of:

- An Assembler
- An Emulator
- A Debugger



# The Assembler:

- Assemble the language described in the ELB816 specification
- Two pass assembler:
  - First pass deals with labels and directives, calculates address of each instruction
  - Second pass converts instructions and arguments to machine code and writes to a hex file

## Mnemonics

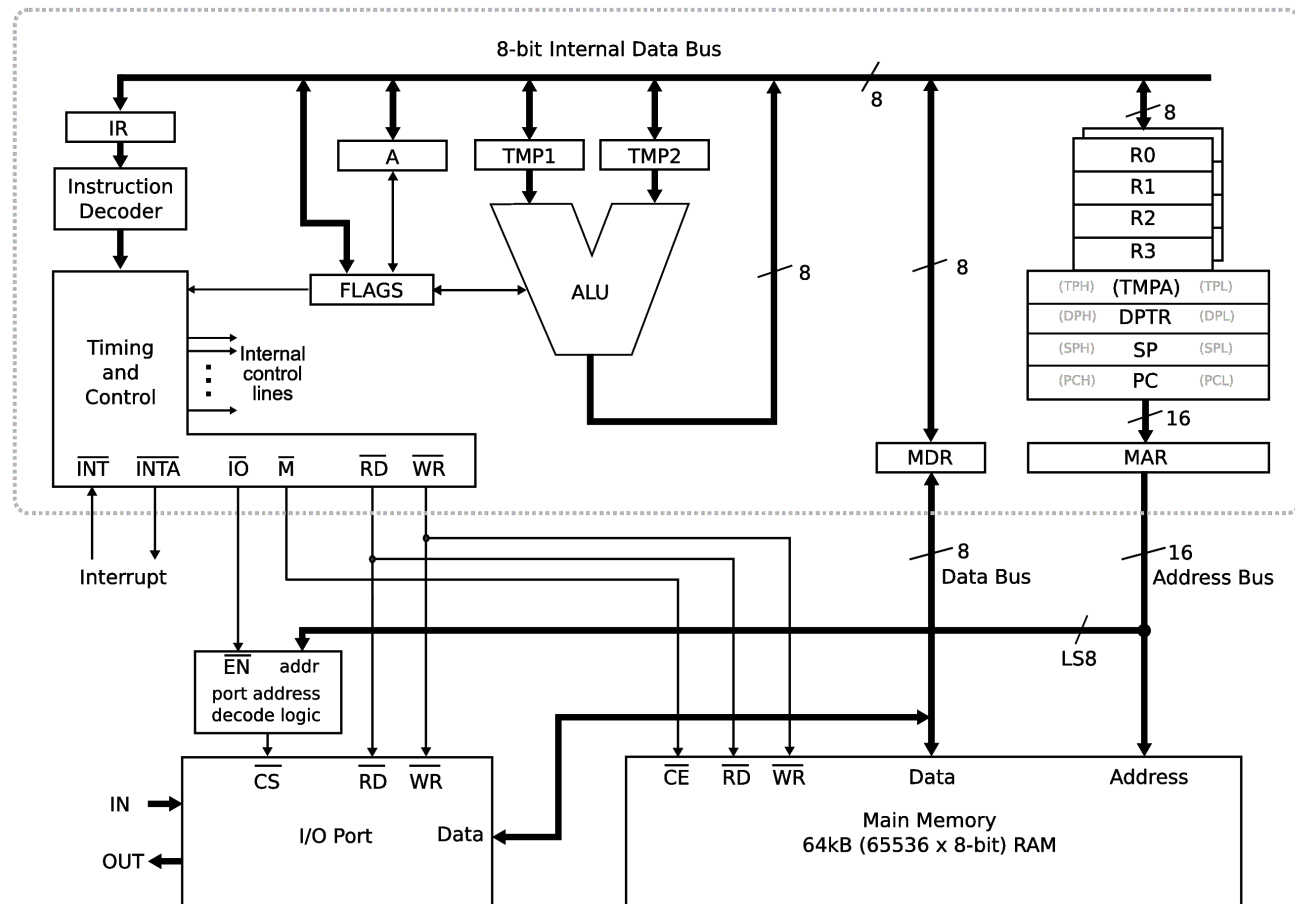
“MOV” | “NOP” | “XCSD” | “SFA” | “LAF” | “ANL”  
| “ORL” | “XRL” | “RL” | “RLC” | “RR” | “RRC”  
| “INC” | “DEC” | “SET” | “CLR” | “CPL” | “ADD”  
| “ADDC” | “SUB” | “SUBB” | “PJUMP” | “PCALL”  
| “LJMP” | “LCALL” | “DJNZ” | “CJNE” | “RET”  
| “RETI” | “SJMP” | “JMP” | “JZ” | “JNZ” | “JC”  
| “JNC” | “JPO” | “JPE” | “JS” | “JNS” | “PUSH”  
| “POP” | “IN” | “OUT” | “HLT”

## Directives

“ORG” | “EQU” | “DB” | “DS” | “CSEG” | “RSEG”  
| “SEGMENT” | “PUBLIC” | “EXTERN”

# The Emulator:

- Will interpret this machine code according to the ELB816 specification
- Will compile for Linux and Intel MCS-51 (8051/2)
- Will communicate with the debugger via a serial interface



# The Debugger:

- Will allow runtime debugging of programs running on the emulator on Linux or an 8051/2
- Will communicate with the emulator via a serial interface

get\_reg() set\_reg() read\_byte() write\_byte() read\_memory()  
write\_memory() read\_port() write\_port() set\_breakpoint()  
step\_in\_to() step\_out\_of() step\_over() etc...