8051 Simulator Layout

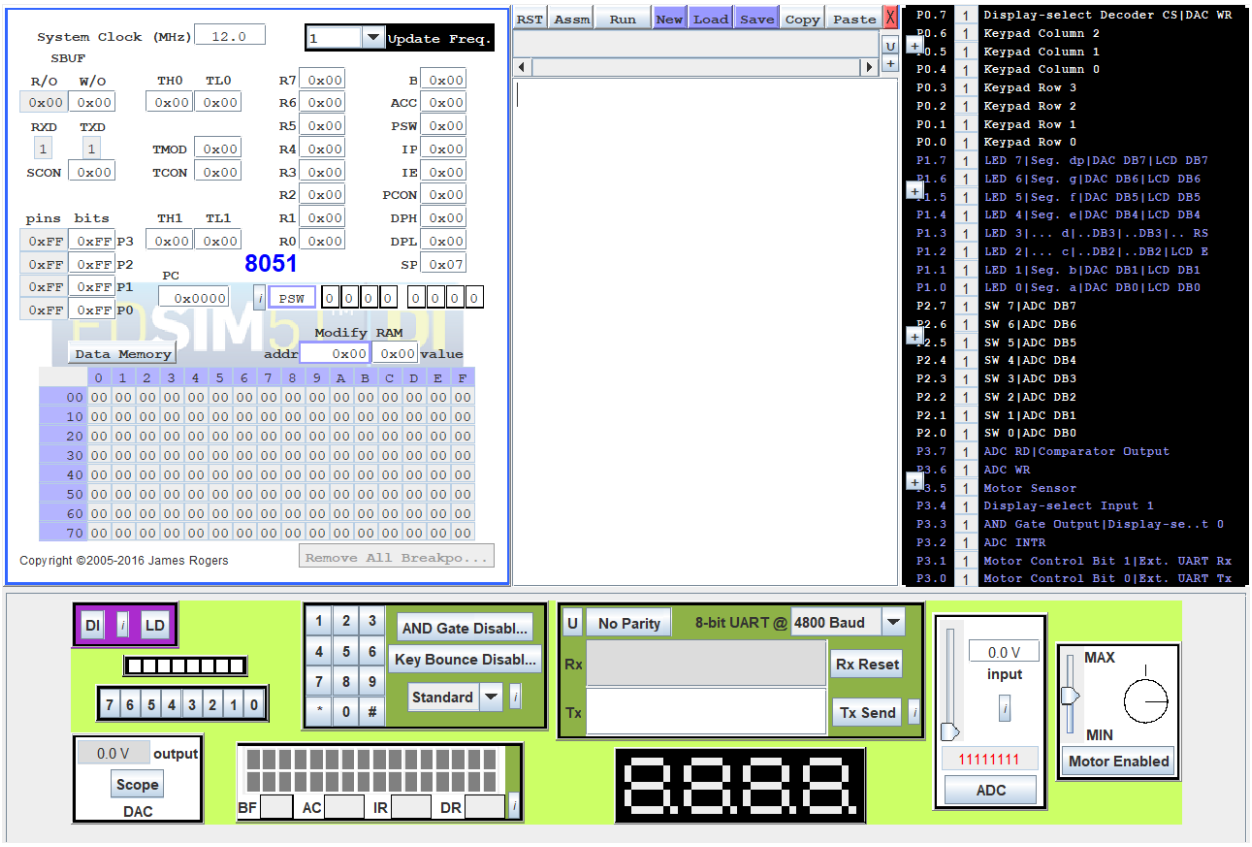


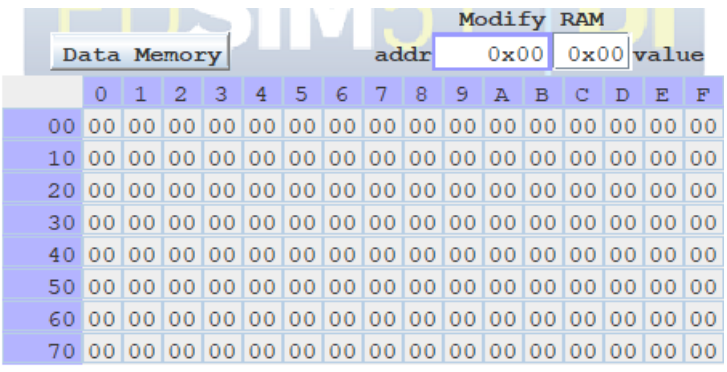
Figure 1 - 8051 Simulator

The simulator window is broken into four components: a memory display (top left), an assembly editor (top middle), a port display (top right), and a display of all the peripherals connected to the 8051 (bottom).

Memory Display

The memory display (top left) is an interactive panel that displays all of the internal memory components of the 8051. Each field can be edited manually. Unfortunately, there is no way to reset their values to zero without closing and reopenning the program.

At the bottom is a table displaying the memory at each address in RAM. There are two parts to it: data and code. Pressing the "Data Memory" / "Code Memory" button will switch between them. To edit the value at a memory address, edit the two fields at the top



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 2 - RAM

right until "Modify Ram". These represent the internal RAM of the 8051.

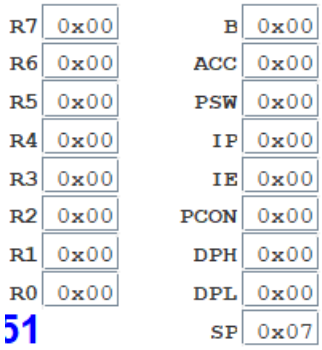


Figure 3 - General & Special Registers

Note: The DPTR register is a 16-bit register and can be addressed as just DPTR. However, DPTR's high and low bytes can also be addressed individually as DPH and DPL.

Around the top right are two columns of fields. The left column represents the general purpose registers, named R0 – R7. The right column represents the special purpose registers. The special purpose registers service specific uses.

Register	Description
B	Only used by the MUL and DIV instructions. Often also used like a 9th "R" register.
ACC	Accumulator – used to hold the cumulative results of a large number of operations. Used by pretty much every instruction.
PSW	Processor Status Word – also called the flag register.
IP	Interrupt Priority – how important the current interrupt is. Priorities determine whether an interrupt is important enough to interrupt another interrupt.
IE	Interrupt Enable – enables or disables an interrupt.
PCON	Power Control – used to force the 8051 into power saving mode. Made from several bit flags.
DPH	Data Pointer High – the high byte of the DPTR register.
DPL	Data Pointer Low – the low byte of the DPTR register.
SP	Stack Pointer – points to the top of the stack.

The PC register (shown to the right) – called the Program Counter – is used to determine the address of the next instruction to execute. Normally, the PC register's value cannot be assigned a value like other registers can. However, this can be done with a jump instruction, which is the equivalent of moving elsewhere in code.

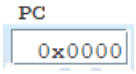


Figure 4 - PC Register

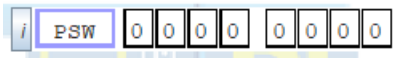


Figure 5 - Bit Addressable Fields

The component (shown to the left) displays the bits within a byte.

The left-most field displays the name of the byte being edited. By default, it is the PSW register, but can be changed to any other register names – such as ACC, B, DPH, etc. – or memory address. If the address if bit-addressable, then it can be edited.

Symbol	Function
CY	Carry flag
AC	Auxilliary carry flag
F0	Available to user for general purpose use
RS1	Register bank selector bit 1
RS0	Register bank selector bit 0
OV	Overflow flag
UD	User definable flag

An an important note, the PSW register can be broken down into 8 individual bits, as shown in the table to the right. The CY, AC, OV, and P bits are conditional flags set by various instructions. RS1 and RS0 are used to determine the current register bank – of which there are four.

P	Parity flag

The other components shown in the memory display will not be discussed in this document.

Editor

The assembly editor serves two main purposes: writing assembly code, and testing it. Assembly code can be entered in the white text area. Above that is a small box with scrollbars that displays error messages when they occur. To actually assemble and run code, click the "Assm" button.