em8051

Jari Komppa

em8051

Jari Komppa

Published 2.4.2006

Table of Contents

Chapter 1. Preface

1.1. What is em8051

em8051 is a software simulator of the intel MCS-51 microcontroller architecture. Its goal is to be a free software simulator capable enough to help development and debugging of 8051-based applications.

While the em8051 executable is useful as is, source code is also provided so that additional features can be added to, for example, simulate different kinds of configurations.

While effort has been made to make this program as flawless as possible, there are bound to be some problems.

Some of the supported features include:

- Full 8051 instruction set
- Timer 0 and timer 1 modes 0,1,2 and 3
- Interrupt priorities
- Debugging exceptions for invalid instructions, odd stack behavior, messing up important registers in interrupts
- Intel HEX file loading
- Single-stepping and various speed run modes, including "real time", if host is powerful enough
- ncurses-based text-mode UI, including
  - Main view, with memory, stack, opcode and disassembly, history view of most registers, cycle- and real-time counters.
  - Memory editor view which shows all five types of memory at the same time
  - 'logic board' view with ports P0-P3 wired to leds and switches, plus optional additional widgets, including 7-segment displays and 8-bit shift registers
  - Options view, where user can disable exceptions and set the desired clock speed

1.2. What em8051 is not

em8051 does not attempt to be a hardware emulator, in a way that no sub-clock actions are emulated. For instance, the `MUL AB` instruction, which takes four cpu cycles (or 48 clock oscillations) executes on the first cycle and then waits for the rest of the operation. This may cause some slightly errornous timing behavior.

1.3. Legalese

1.3.1. MIT license

Chapter 2. Using em8051

2.1. Running em8051

em8051 can be launched either without any commandline parameters, or optionally with the name of an Intel HEX-format object file.

2.2. Global keys

The simulator has several global keys that work regardless of the viewing mode.

- Function keys

  Quick switch between views.

- v

  Cycle between views

- k

  Set breakpoint. When the simulation reaches the breakpoint, the simulation is stopped. Pressing k again clears the breakpoint.

- g

  Set program counter value.

- h

  Shows brief help.

- l

  Load an intel HEX format .obj file.

- space

  Simulation step. Depending on the options, this steps a single cpu cycle or a single instruction. An instruction may take more than one cpu cycle.

- r

  Toggle run mode. In run mode, the simulator executes cpu cycles automatically at desired speed. Run mode is also terminated if space is pressed or some popup event, such as exception, occurs.

- + and -

  Adjust run speed. Various run speed modes are supported.

- Home key

  Reset emulator. Shows popup giving user choise of reset style, ranging from simply setting program counter to zero to a complete wipe of all memory.

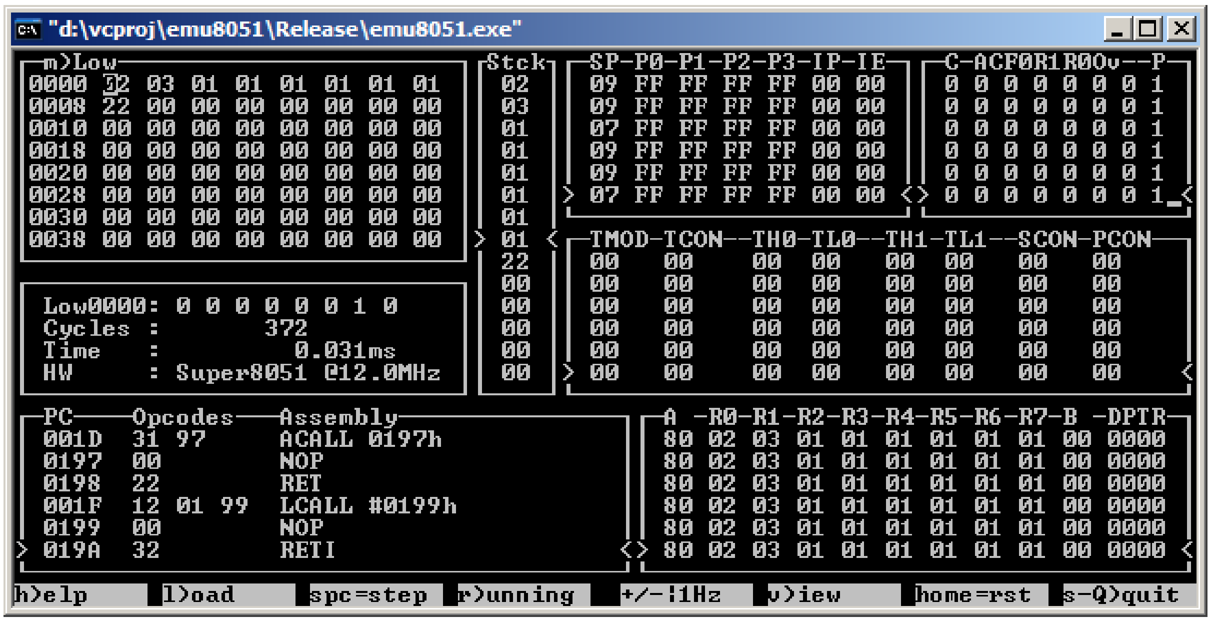
- End key

  Resets clock counter, useful for timing events, for example from one breakpoint to the next.

- Shift-Q

  Quit em8051

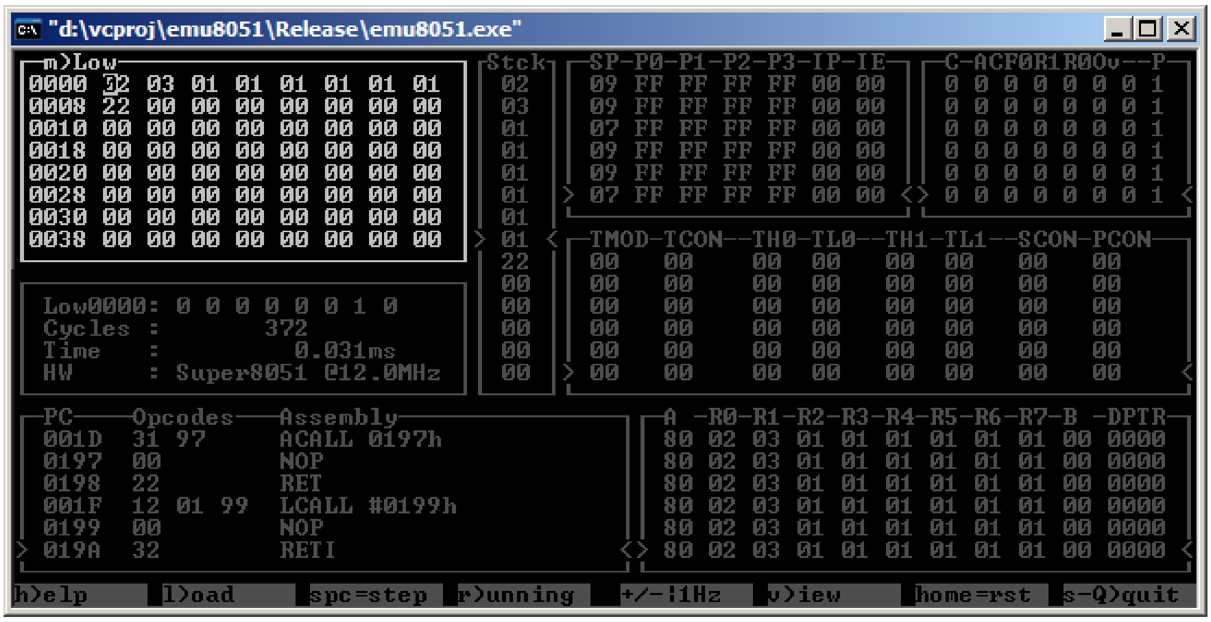
2.3. Main view



```
"d:\vcproj\emu8051\Release\emu8051.exe"
┌m)Low──────────────────────────┐┌Stck┐┌SP-P0-P1-P2-P3-IP-IE─┐┌C-ACF0R1R0Ov--P─┐
│0000 02 03 01 01 01 01 01 01   ││ 02 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0008 22 00 00 00 00 00 00 00   ││ 03 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0010 00 00 00 00 00 00 00 00   ││ 01 ││ 07 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0018 00 00 00 00 00 00 00 00   ││ 01 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0020 00 00 00 00 00 00 00 00   ││ 01 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0028 00 00 00 00 00 00 00 00   ││ 01 │> 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1_<
│0030 00 00 00 00 00 00 00 00   ││ 01 │└─────────────────────┘└─────────────────┘
│0038 00 00 00 00 00 00 00 00   │> 01 <┌TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON──┐
└───────────────────────────────┘│ 22 ││ 00   00    00  00   00  00   00   00    │
┌───────────────────────────────┐│ 00 ││ 00   00    00  00   00  00   00   00    │
│ Low0000: 0 0 0 0 0 0 1 0      ││ 00 ││ 00   00    00  00   00  00   00   00    │
│ Cycles :        372           ││ 00 ││ 00   00    00  00   00  00   00   00    │
│ Time   :          0.031ms     ││ 00 ││ 00   00    00  00   00  00   00   00    │
│ HW     : Super8051 @12.0MHz   ││ 00 │> 00   00    00  00   00  00   00   00   <
└───────────────────────────────┘└────┘└────────────────────────────────────────┘
┌PC─────Opcodes───Assembly──────────────┐┌A -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR─┐
│ 001D  31 97     ACALL 0197h           ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 0197  00        NOP                   ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 0198  22        RET                   ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 001F  12 01 99  LCALL #0199h          ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 0199  00        NOP                   ││ 80 02 03 01 01 01 01 01 01 00 0000 │
> 019A  32        RETI                  <> 80 02 03 01 01 01 01 01 01 00 0000 <
h)elp    l)oad    spc=step  r)unning  +/-¦1Hz   v)iew    home=rst  s-Q)quit
```

The main view

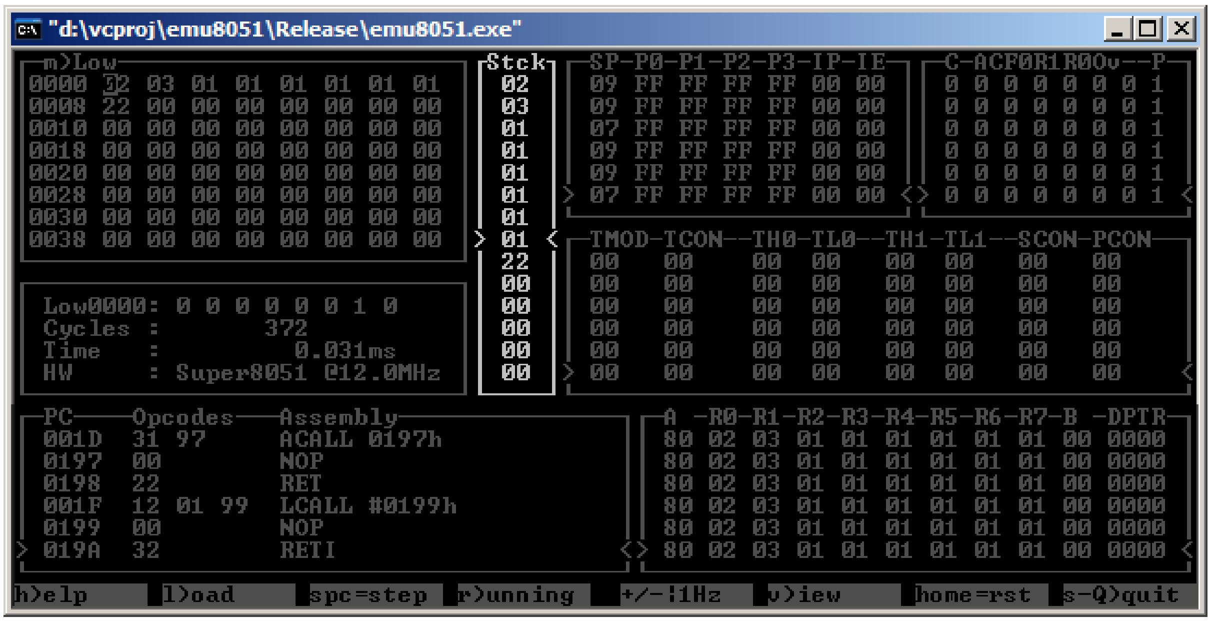
The main view is designed to give as much information about the emulation state as possible. Viewed information includes:



```
"d:\vcproj\emu8051\Release\emu8051.exe"
┌m)Low──────────────────────────┐┌Stck┐┌SP-P0-P1-P2-P3-IP-IE─┐┌C-ACF0R1R0Ov--P─┐
│0000 02 03 01 01 01 01 01 01   ││ 02 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0008 22 00 00 00 00 00 00 00   ││ 03 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0010 00 00 00 00 00 00 00 00   ││ 01 ││ 07 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0018 00 00 00 00 00 00 00 00   ││ 01 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0020 00 00 00 00 00 00 00 00   ││ 01 ││ 09 FF FF FF FF 00 00 ││ 0 0 0 0 0 0 0 1 │
│0028 00 00 00 00 00 00 00 00   ││ 01 │> 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
│0030 00 00 00 00 00 00 00 00   ││ 01 │└─────────────────────┘└─────────────────┘
│0038 00 00 00 00 00 00 00 00   │> 01 <┌TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON──┐
└───────────────────────────────┘│ 22 ││ 00   00    00  00   00  00   00   00    │
┌───────────────────────────────┐│ 00 ││ 00   00    00  00   00  00   00   00    │
│ Low0000: 0 0 0 0 0 0 1 0      ││ 00 ││ 00   00    00  00   00  00   00   00    │
│ Cycles :        372           ││ 00 ││ 00   00    00  00   00  00   00   00    │
│ Time   :          0.031ms     ││ 00 ││ 00   00    00  00   00  00   00   00    │
│ HW     : Super8051 @12.0MHz   ││ 00 │> 00   00    00  00   00  00   00   00   <
└───────────────────────────────┘└────┘└────────────────────────────────────────┘
┌PC─────Opcodes───Assembly──────────────┐┌A -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR─┐
│ 001D  31 97     ACALL 0197h           ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 0197  00        NOP                   ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 0198  22        RET                   ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 001F  12 01 99  LCALL #0199h          ││ 80 02 03 01 01 01 01 01 01 00 0000 │
│ 0199  00        NOP                   ││ 80 02 03 01 01 01 01 01 01 00 0000 │
> 019A  32        RETI                  <> 80 02 03 01 01 01 01 01 01 00 0000 <
h)elp    l)oad    spc=step  r)unning  +/-¦1Hz   v)iew    home=rst  s-Q)quit
```

Memory editor

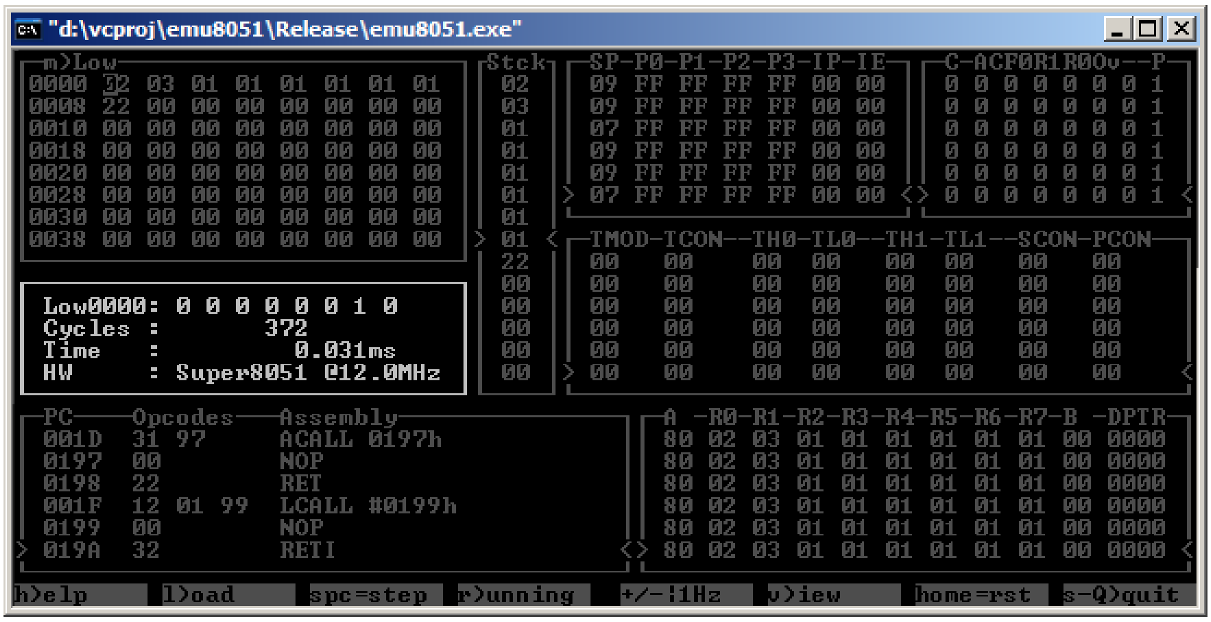
Memory view / editor. Shows 64 bytes of memory at once, and can be toggled between the five memory types of the 8051 using the b key. Cursor keys, pageup and pagedown can be used to move cursor in the memory view, and numbers plus a-f keys can be used to adjust memory values.



```
"d:\vcproj\emu8051\Release\emu8051.exe"
 m)Low                                Stck  SP-P0-P1-P2-P3-IP-IE    C-ACF0R1R0Ov--P
 0000 32 03 01 01 01 01 01 01          02   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0008 22 00 00 00 00 00 00 00          03   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0010 00 00 00 00 00 00 00 00          01   07 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0018 00 00 00 00 00 00 00 00          01   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0020 00 00 00 00 00 00 00 00          01   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0028 00 00 00 00 00 00 00 00          01 > 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
 0030 00 00 00 00 00 00 00 00          01
 0038 00 00 00 00 00 00 00 00        > 01 < TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON
                                       22   00   00    00  00   00  00   00   00
                                       00   00   00    00  00   00  00   00   00
  Low0000: 0 0 0 0 0 0 1 0             00   00   00    00  00   00  00   00   00
  Cycles  :          372               00   00   00    00  00   00  00   00   00
  Time    :            0.031ms         00   00   00    00  00   00  00   00   00
  HW      : Super8051 @12.0MHz         00 > 00   00    00  00   00  00   00   00   <

  PC    Opcodes   Assembly                       A  -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR
  001D  31 97     ACALL 0197h                    80 02 03 01 01 01 01 01 01 00 0000
  0197  00        NOP                            80 02 03 01 01 01 01 01 01 00 0000
  0198  22        RET                            80 02 03 01 01 01 01 01 01 00 0000
  001F  12 01 99  LCALL #0199h                   80 02 03 01 01 01 01 01 01 00 0000
  0199  00        NOP                            80 02 03 01 01 01 01 01 01 00 0000
> 019A  32        RETI                         <> 80 02 03 01 01 01 01 01 01 00 0000 <
h)elp      l)oad      spc=step  r)unning  +/-:1Hz  v)iew      home=rst  s-Q)quit
```

Stack view

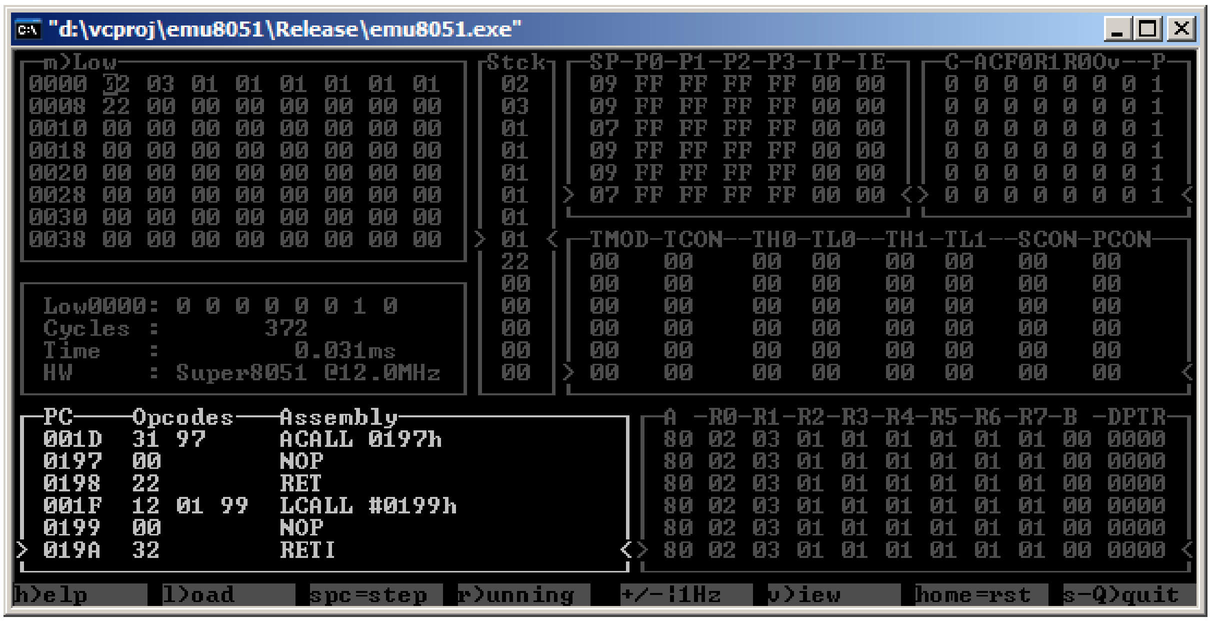
Stack view shows the current state of the stack.



```
"d:\vcproj\emu8051\Release\emu8051.exe"
 m)Low                                Stck  SP-P0-P1-P2-P3-IP-IE    C-ACF0R1R0Ov--P
 0000 32 03 01 01 01 01 01 01          02   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0008 22 00 00 00 00 00 00 00          03   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0010 00 00 00 00 00 00 00 00          01   07 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0018 00 00 00 00 00 00 00 00          01   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0020 00 00 00 00 00 00 00 00          01   09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0028 00 00 00 00 00 00 00 00          01 > 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
 0030 00 00 00 00 00 00 00 00          01
 0038 00 00 00 00 00 00 00 00        > 01 < TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON
                                       22   00   00    00  00   00  00   00   00
                                       00   00   00    00  00   00  00   00   00
  Low0000: 0 0 0 0 0 0 1 0             00   00   00    00  00   00  00   00   00
  Cycles  :          372               00   00   00    00  00   00  00   00   00
  Time    :            0.031ms         00   00   00    00  00   00  00   00   00
  HW      : Super8051 @12.0MHz         00 > 00   00    00  00   00  00   00   00   <

  PC    Opcodes   Assembly                       A  -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR
  001D  31 97     ACALL 0197h                    80 02 03 01 01 01 01 01 01 00 0000
  0197  00        NOP                            80 02 03 01 01 01 01 01 01 00 0000
  0198  22        RET                            80 02 03 01 01 01 01 01 01 00 0000
  001F  12 01 99  LCALL #0199h                   80 02 03 01 01 01 01 01 01 00 0000
  0199  00        NOP                            80 02 03 01 01 01 01 01 01 00 0000
> 019A  32        RETI                         <> 80 02 03 01 01 01 01 01 01 00 0000 <
h)elp      l)oad      spc=step  r)unning  +/-:1Hz  v)iew      home=rst  s-Q)quit
```

Misc view

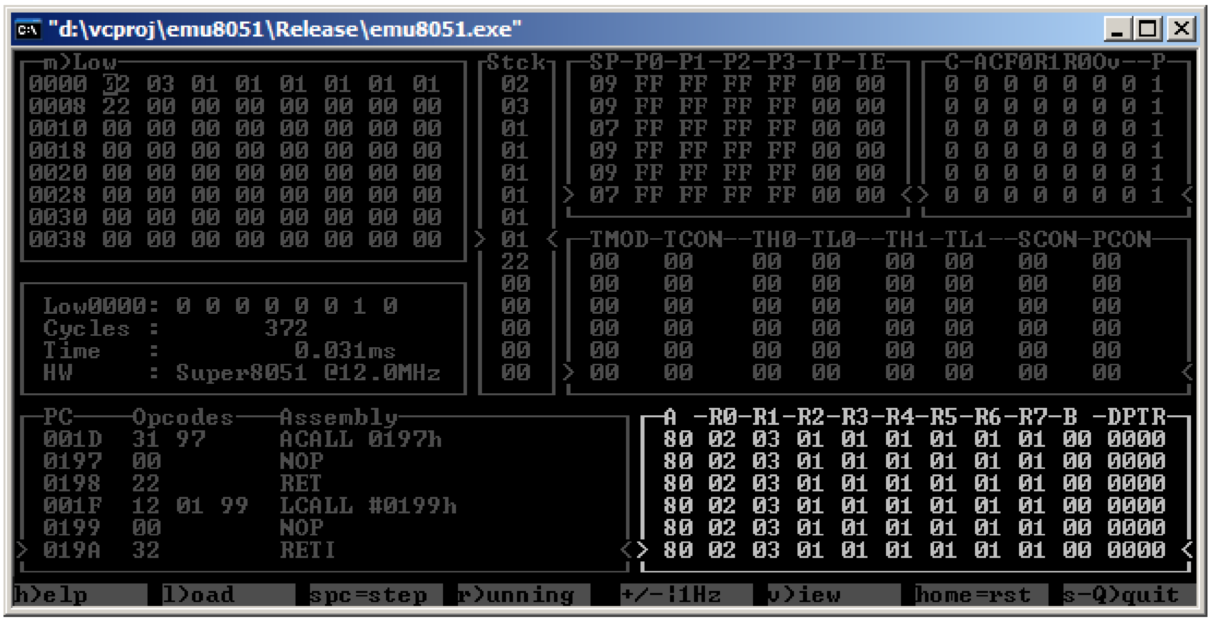
Misc. view, below the memory view shows the address and bit mask of the currently edited byte. Apart from this, the elapsed clock cycles and real time are shown, along with the currently emulated hardware and clock speed.



```
"d:\vcproj\emu8051\Release\emu8051.exe"
m)Low                                      Stck  SP-P0-P1-P2-P3-IP-IE     C-ACF0R1R0Ov--P
0000 02 03 01 01 01 01 01 01                02   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0008 22 00 00 00 00 00 00 00                03   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0010 00 00 00 00 00 00 00 00                01   07 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0018 00 00 00 00 00 00 00 00                01   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0020 00 00 00 00 00 00 00 00                01   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0028 00 00 00 00 00 00 00 00                01 > 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
0030 00 00 00 00 00 00 00 00                01
0038 00 00 00 00 00 00 00 00              > 01 < TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON
                                            22   00   00    00  00   00  00   00   00
 Low0000: 0 0 0 0 0 0 1 0                   00   00   00    00  00   00  00   00   00
 Cycles :         372                       00   00   00    00  00   00  00   00   00
 Time   :           0.031ms                 00   00   00    00  00   00  00   00   00
 HW     : Super8051 @12.0MHz                00   00   00    00  00   00  00   00   00
                                            00 > 00   00    00  00   00  00   00   00 <
 PC    Opcodes   Assembly                      A  -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR
 001D  31 97     ACALL 0197h                   80 02 03 01 01 01 01 01 01 00 0000
 0197  00        NOP                           80 02 03 01 01 01 01 01 01 00 0000
 0198  22        RET                           80 02 03 01 01 01 01 01 01 00 0000
 001F  12 01 99  LCALL #0199h                  80 02 03 01 01 01 01 01 01 00 0000
 0199  00        NOP                           80 02 03 01 01 01 01 01 01 00 0000
> 019A  32        RETI                        <> 80 02 03 01 01 01 01 01 01 00 0000 <
h)elp    l)oad    spc=step  r)unning  +/-:1Hz  v)iew    home=rst  s-Q)quit
```

Execution window

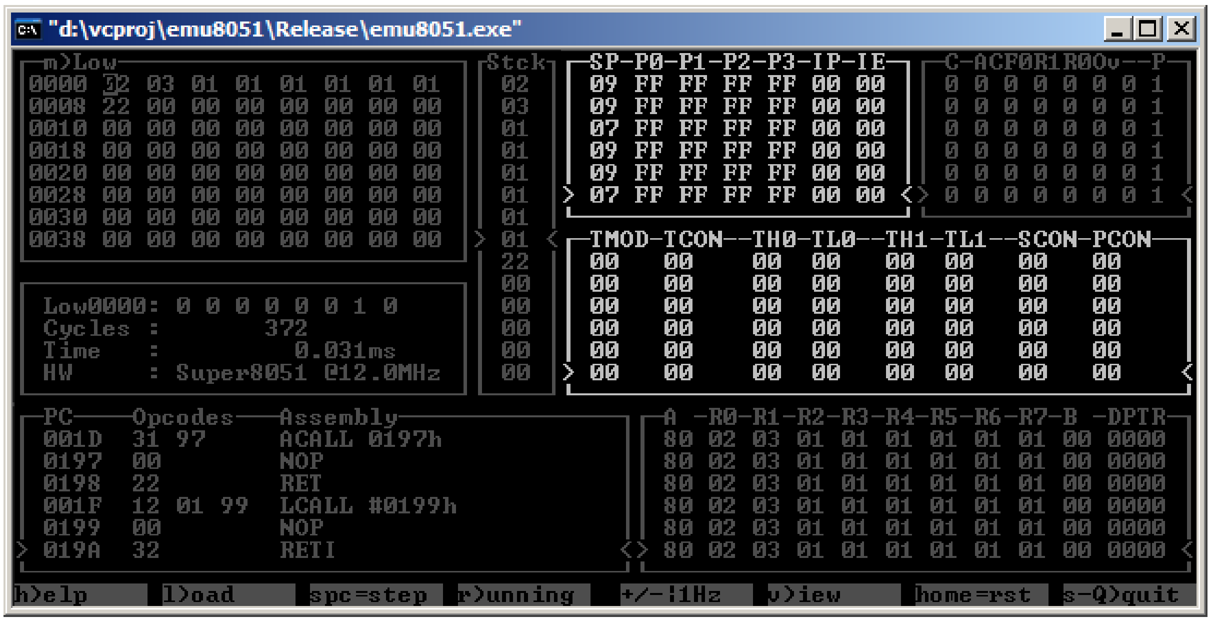
On lower left is the execution window, which shows the program counter, opcodes and disassembly. The lowest line shows the most recently executed instruction.



```
"d:\vcproj\emu8051\Release\emu8051.exe"
m)Low                                      Stck  SP-P0-P1-P2-P3-IP-IE     C-ACF0R1R0Ov--P
0000 02 03 01 01 01 01 01 01                02   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0008 22 00 00 00 00 00 00 00                03   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0010 00 00 00 00 00 00 00 00                01   07 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0018 00 00 00 00 00 00 00 00                01   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0020 00 00 00 00 00 00 00 00                01   09 FF FF FF FF 00 00     0 0 0 0 0 0 0 1
0028 00 00 00 00 00 00 00 00                01 > 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
0030 00 00 00 00 00 00 00 00                01
0038 00 00 00 00 00 00 00 00              > 01 < TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON
                                            22   00   00    00  00   00  00   00   00
 Low0000: 0 0 0 0 0 0 1 0                   00   00   00    00  00   00  00   00   00
 Cycles :         372                       00   00   00    00  00   00  00   00   00
 Time   :           0.031ms                 00   00   00    00  00   00  00   00   00
 HW     : Super8051 @12.0MHz                00   00   00    00  00   00  00   00   00
                                            00 > 00   00    00  00   00  00   00   00 <
 PC    Opcodes   Assembly                      A  -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR
 001D  31 97     ACALL 0197h                   80 02 03 01 01 01 01 01 01 00 0000
 0197  00        NOP                           80 02 03 01 01 01 01 01 01 00 0000
 0198  22        RET                           80 02 03 01 01 01 01 01 01 00 0000
 001F  12 01 99  LCALL #0199h                  80 02 03 01 01 01 01 01 01 00 0000
 0199  00        NOP                           80 02 03 01 01 01 01 01 01 00 0000
> 019A  32        RETI                        <> 80 02 03 01 01 01 01 01 01 00 0000 <
h)elp    l)oad    spc=step  r)unning  +/-:1Hz  v)iew    home=rst  s-Q)quit
```

Common registers

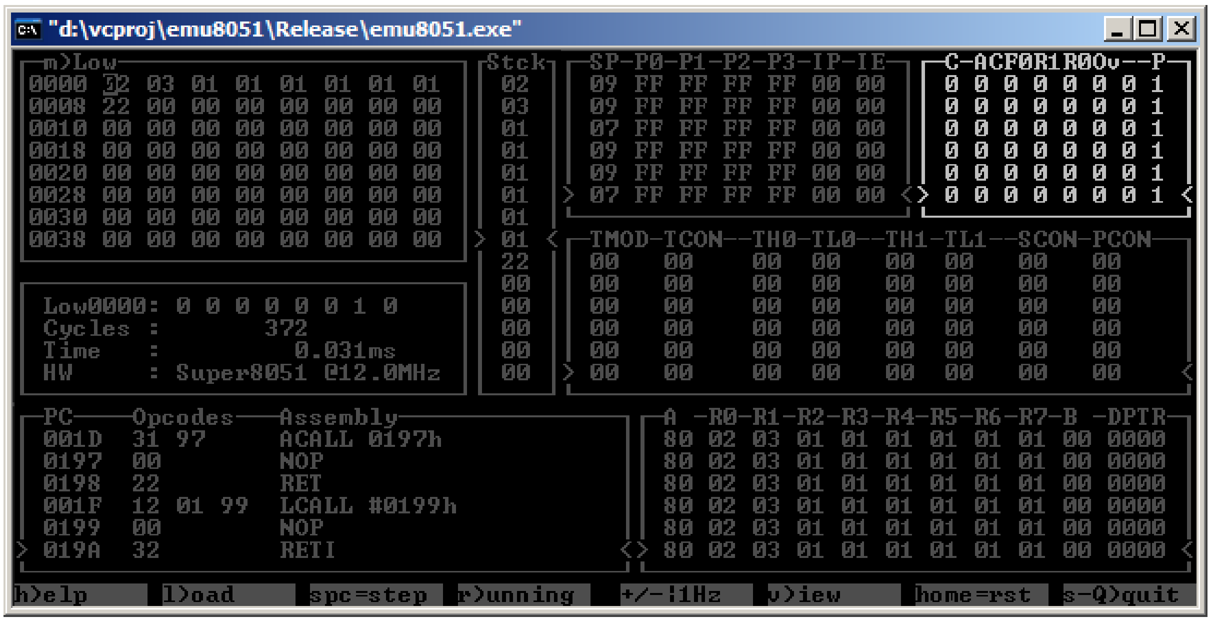
On the right from the execution window are the common registers. User can switch focus between the memory editor and the common registers with the tab key, and edit the common registers directly.



```
"d:\vcproj\emu8051\Release\emu8051.exe"
 m)Low                                   Stck  SP-P0-P1-P2-P3-IP-IE    C-ACF0R1R0Ov--P
 0000 02 03 01 01 01 01 01 01             02    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0008 22 00 00 00 00 00 00 00             03    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0010 00 00 00 00 00 00 00 00             01    07 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0018 00 00 00 00 00 00 00 00             01    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0020 00 00 00 00 00 00 00 00             01    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0028 00 00 00 00 00 00 00 00             01  > 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
 0030 00 00 00 00 00 00 00 00             01
 0038 00 00 00 00 00 00 00 00           > 01 <  TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON
                                          22    00   00    00  00   00  00   00   00
                                          00    00   00    00  00   00  00   00   00
  Low0000: 0 0 0 0 0 0 1 0                00    00   00    00  00   00  00   00   00
  Cycles :        372                     00    00   00    00  00   00  00   00   00
  Time   :          0.031ms               00    00   00    00  00   00  00   00   00
  HW     : Super8051 @12.0MHz             00  > 00   00    00  00   00  00   00   00   <

 PC     Opcodes   Assembly                          A -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR
  001D   31 97     ACALL 0197h                      80 02 03 01 01 01 01 01 01 00 0000
  0197   00        NOP                              80 02 03 01 01 01 01 01 01 00 0000
  0198   22        RET                              80 02 03 01 01 01 01 01 01 00 0000
  001F   12 01 99  LCALL #0199h                     80 02 03 01 01 01 01 01 01 00 0000
  0199   00        NOP                              80 02 03 01 01 01 01 01 01 00 0000
> 019A   32        RETI                          <> 80 02 03 01 01 01 01 01 01 00 0000 <
h>elp    l>oad    spc=step  r>unning  +/-:1Hz  v>iew    home=rst  s-Q>quit
```

Control registers and output ports

Two of the remaining windows show the state of the control registers and output ports.

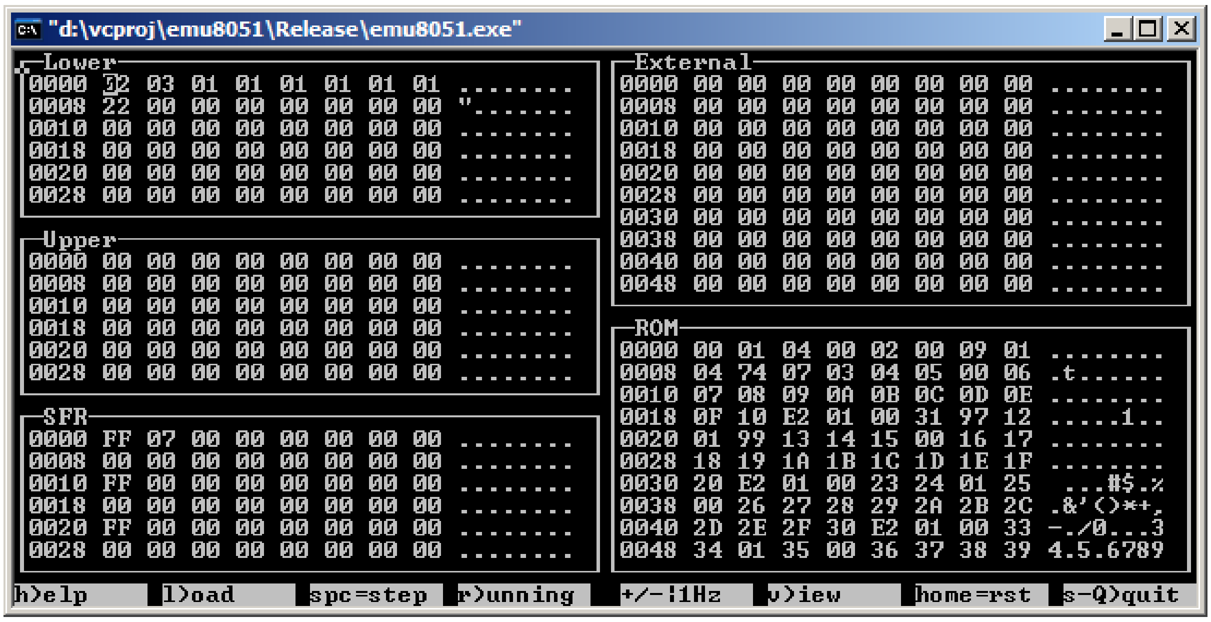


```
"d:\vcproj\emu8051\Release\emu8051.exe"
 m)Low                                   Stck  SP-P0-P1-P2-P3-IP-IE    C-ACF0R1R0Ov--P
 0000 02 03 01 01 01 01 01 01             02    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0008 22 00 00 00 00 00 00 00             03    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0010 00 00 00 00 00 00 00 00             01    07 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0018 00 00 00 00 00 00 00 00             01    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0020 00 00 00 00 00 00 00 00             01    09 FF FF FF FF 00 00    0 0 0 0 0 0 0 1
 0028 00 00 00 00 00 00 00 00             01  > 07 FF FF FF FF 00 00 <> 0 0 0 0 0 0 0 1 <
 0030 00 00 00 00 00 00 00 00             01
 0038 00 00 00 00 00 00 00 00           > 01 <  TMOD-TCON--TH0-TL0--TH1-TL1--SCON-PCON
                                          22    00   00    00  00   00  00   00   00
                                          00    00   00    00  00   00  00   00   00
  Low0000: 0 0 0 0 0 0 1 0                00    00   00    00  00   00  00   00   00
  Cycles :        372                     00    00   00    00  00   00  00   00   00
  Time   :          0.031ms               00    00   00    00  00   00  00   00   00
  HW     : Super8051 @12.0MHz             00  > 00   00    00  00   00  00   00   00   <

 PC     Opcodes   Assembly                          A -R0-R1-R2-R3-R4-R5-R6-R7-B -DPTR
  001D   31 97     ACALL 0197h                      80 02 03 01 01 01 01 01 01 00 0000
  0197   00        NOP                              80 02 03 01 01 01 01 01 01 00 0000
  0198   22        RET                              80 02 03 01 01 01 01 01 01 00 0000
  001F   12 01 99  LCALL #0199h                     80 02 03 01 01 01 01 01 01 00 0000
  0199   00        NOP                              80 02 03 01 01 01 01 01 01 00 0000
> 019A   32        RETI                          <> 80 02 03 01 01 01 01 01 01 00 0000 <
h>elp    l>oad    spc=step  r>unning  +/-:1Hz  v>iew    home=rst  s-Q>quit
```

Processor status word

The window in the top right corner shows the state of the processor status word bits.

2.4. Memory editor view



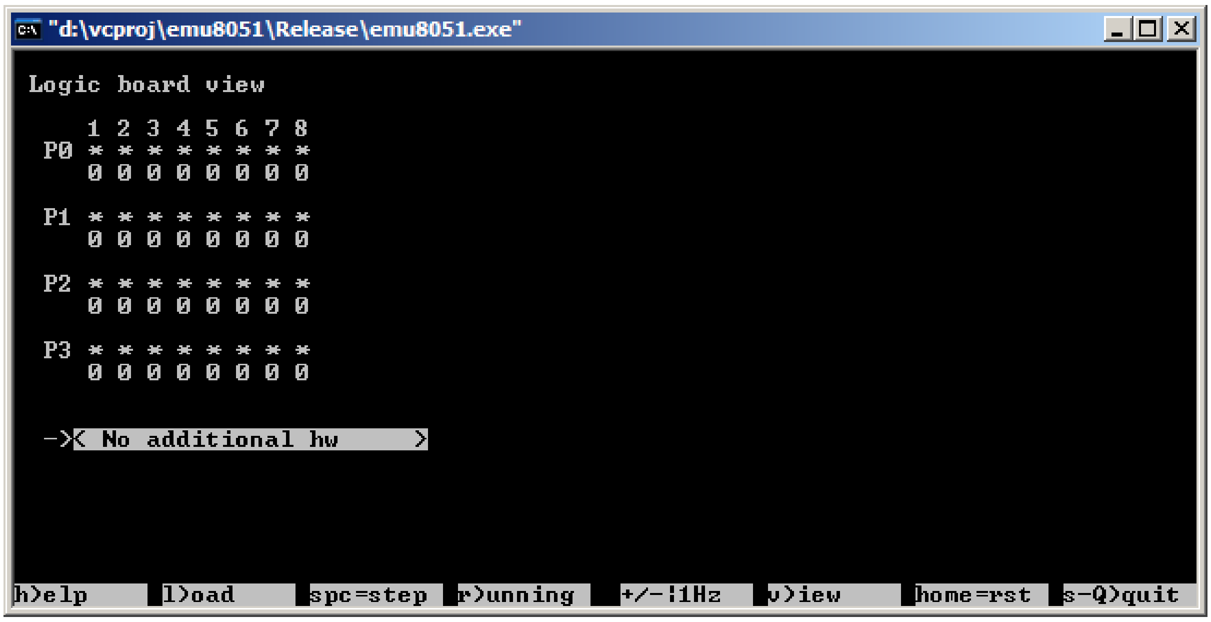
```
"d:\vcproj\emu8051\Release\emu8051.exe"
-Lower-------------------------------------------   -External----------------------------------------
0000 02 03 01 01 01 01 01 01 ........             0000 00 00 00 00 00 00 00 00 ........
0008 22 00 00 00 00 00 00 00 ".......             0008 00 00 00 00 00 00 00 00 ........
0010 00 00 00 00 00 00 00 00 ........             0010 00 00 00 00 00 00 00 00 ........
0018 00 00 00 00 00 00 00 00 ........             0018 00 00 00 00 00 00 00 00 ........
0020 00 00 00 00 00 00 00 00 ........             0020 00 00 00 00 00 00 00 00 ........
0028 00 00 00 00 00 00 00 00 ........             0028 00 00 00 00 00 00 00 00 ........
                                                  0030 00 00 00 00 00 00 00 00 ........
-Upper-------------------------------------------   0038 00 00 00 00 00 00 00 00 ........
0000 00 00 00 00 00 00 00 00 ........             0040 00 00 00 00 00 00 00 00 ........
0008 00 00 00 00 00 00 00 00 ........             0048 00 00 00 00 00 00 00 00 ........
0010 00 00 00 00 00 00 00 00 ........
0018 00 00 00 00 00 00 00 00 ........             -ROM----------------------------------------------
0020 00 00 00 00 00 00 00 00 ........             0000 00 01 04 00 02 00 09 01 ........
0028 00 00 00 00 00 00 00 00 ........             0008 04 74 07 03 04 05 00 06 .t......
                                                  0010 07 08 09 0A 0B 0C 0D 0E ........
-SFR---------------------------------------------   0018 0F 10 E2 01 00 31 97 12 .....1..
0000 FF 07 00 00 00 00 00 00 ........             0020 01 99 13 14 15 00 16 17 ........
0008 00 00 00 00 00 00 00 00 ........             0028 18 19 1A 1B 1C 1D 1E 1F ........
0010 FF 00 00 00 00 00 00 00 ........             0030 20 E2 01 00 23 24 01 25  ...#$.%
0018 00 00 00 00 00 00 00 00 ........             0038 00 26 27 28 29 2A 2B 2C .&'()*+,
0020 FF 00 00 00 00 00 00 00 ........             0040 2D 2E 2F 30 E2 01 00 33 -./0...3
0028 00 00 00 00 00 00 00 00 ........             0048 34 01 35 00 36 37 38 39 4.5.6789
h)elp      l)oad      spc=step  r)unning   +/-:1Hz    v)iew      home=rst  s-Q)quit
```

Memory editor view

The second view shows all of the five memory areas at the same time. User can switch between the memory areas using the tab key.

The editors work the same way as with the main view memory editors, with the additional feature of showing the current ASCII value of the contained bytes.

2.5. Logic board view



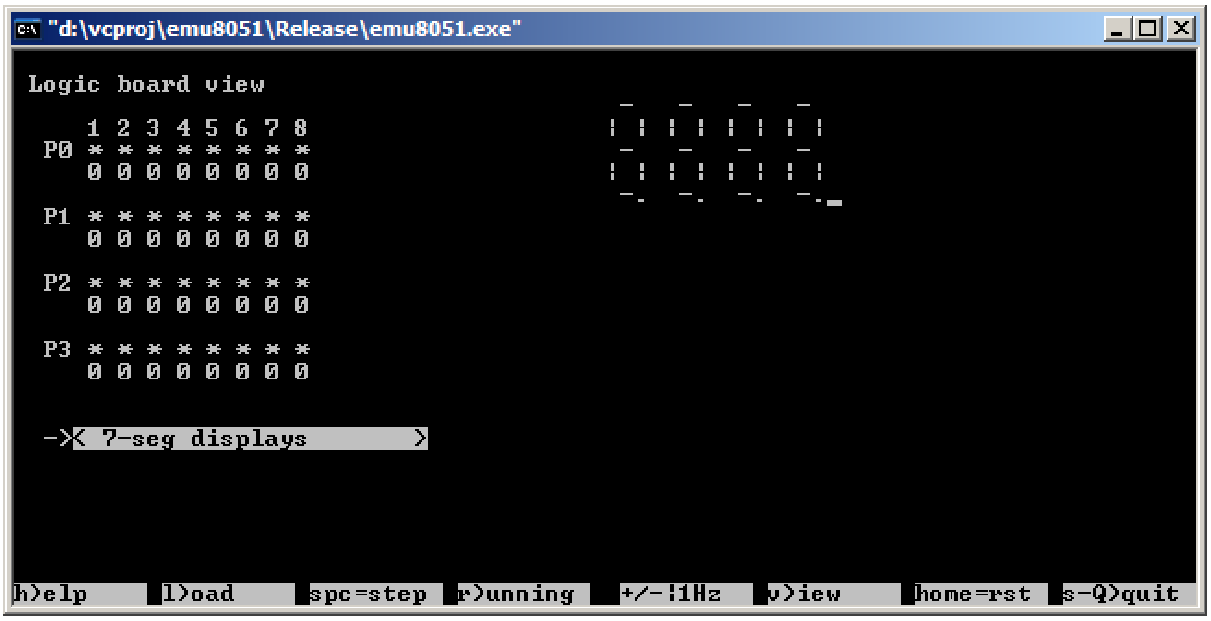
```
"d:\vcproj\emu8051\Release\emu8051.exe"

Logic board view

    1 2 3 4 5 6 7 8
 P0 * * * * * * * *
    0 0 0 0 0 0 0 0

 P1 * * * * * * * *
    0 0 0 0 0 0 0 0

 P2 * * * * * * * *
    0 0 0 0 0 0 0 0

 P3 * * * * * * * *
    0 0 0 0 0 0 0 0


 ->< No additional hw    >





h)elp      l)oad      spc=step  r)unning   +/-:1Hz    v)iew      home=rst  s-Q)quit
```

Logic board view

The logic board view shows the output port bits as LEDs and input ports as switches.

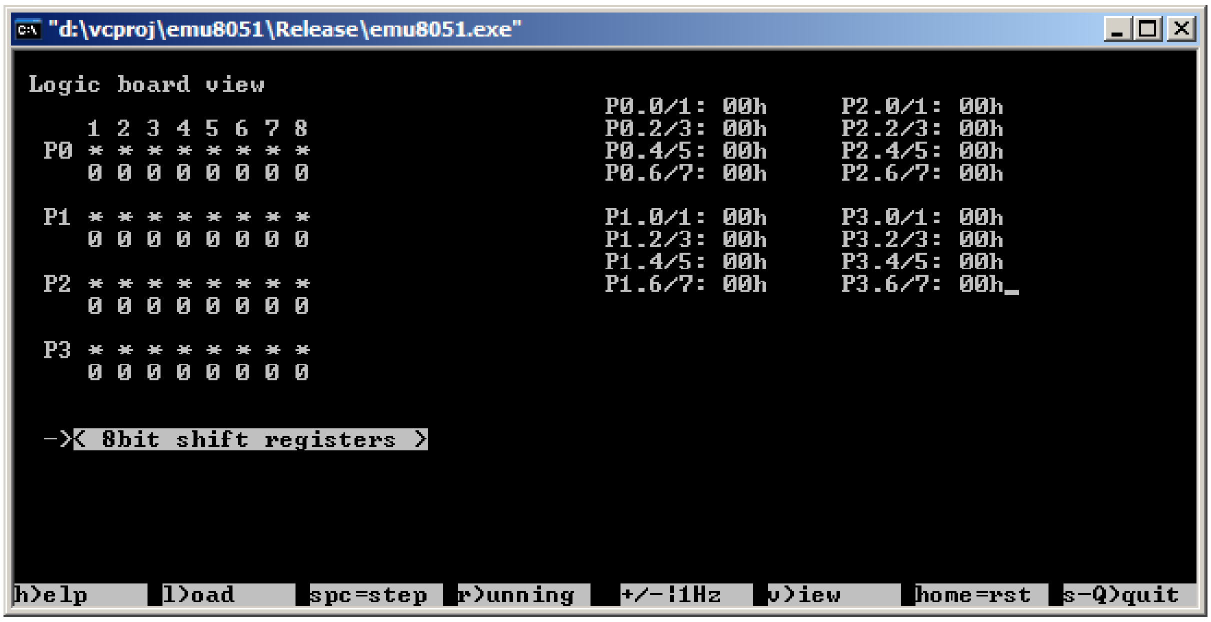
The currently active port can be changed using the up and down cursor keys. The switches are toggled using the keys 1,2,3,4,5,6,7 and 8.

Additionally, different kinds of widgets, such as 7-segment displays or 8-bit shift registers can be enabled by using the left and right keys on the additional hardware selector.



```
"d:\vcproj\emu8051\Release\emu8051.exe"

 Logic board view
                                                 _   _   _   _
     1 2 3 4 5 6 7 8                            | | | | | | | |
  P0 * * * * * * * *                             _   _   _   _
     0 0 0 0 0 0 0 0                            | | | | | | | |
                                                 _.  _.  _.  _._
  P1 * * * * * * * *
     0 0 0 0 0 0 0 0

  P2 * * * * * * * *
     0 0 0 0 0 0 0 0

  P3 * * * * * * * *
     0 0 0 0 0 0 0 0

  ->< 7-seg displays     >

h)elp     l)oad     spc=step  r)unning   +/-:1Hz   v)iew     home=rst  s-Q)quit
```

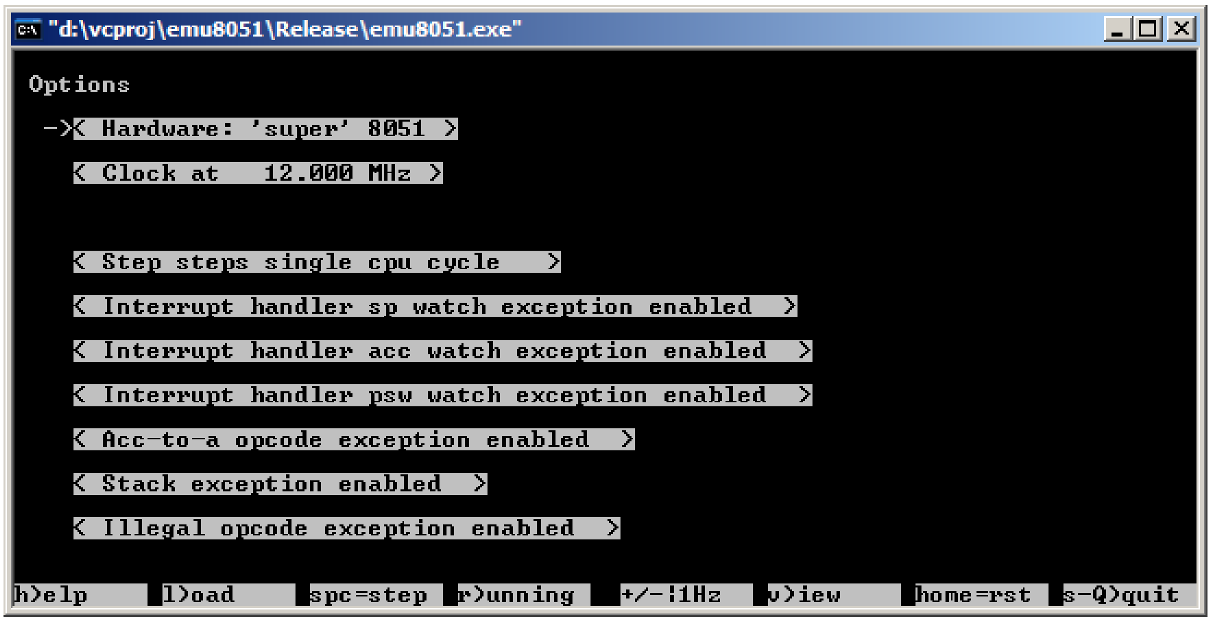
7-segment displays



```
"d:\vcproj\emu8051\Release\emu8051.exe"

 Logic board view
                                           P0.0/1: 00h     P2.0/1: 00h
     1 2 3 4 5 6 7 8                       P0.2/3: 00h     P2.2/3: 00h
  P0 * * * * * * * *                       P0.4/5: 00h     P2.4/5: 00h
     0 0 0 0 0 0 0 0                       P0.6/7: 00h     P2.6/7: 00h

  P1 * * * * * * * *                       P1.0/1: 00h     P3.0/1: 00h
     0 0 0 0 0 0 0 0                       P1.2/3: 00h     P3.2/3: 00h
                                           P1.4/5: 00h     P3.4/5: 00h
  P2 * * * * * * * *                       P1.6/7: 00h     P3.6/7: 00h_
     0 0 0 0 0 0 0 0

  P3 * * * * * * * *
     0 0 0 0 0 0 0 0

  ->< 8bit shift registers >

h)elp     l)oad     spc=step  r)unning   +/-:1Hz   v)iew     home=rst  s-Q)quit
```

8-bit shift registers

2.6. Options view



```
"d:\vcproj\emu8051\Release\emu8051.exe"

Options
  ->< Hardware: 'super' 8051 >
    < Clock at   12.000 MHz >


    < Step steps single cpu cycle   >
    < Interrupt handler sp watch exception enabled  >
    < Interrupt handler acc watch exception enabled  >
    < Interrupt handler psw watch exception enabled  >
    < Acc-to-a opcode exception enabled  >
    < Stack exception enabled  >
    < Illegal opcode exception enabled  >

h>elp     l>oad     spc=step  r>unning  +/-:1Hz   v>iew     home=rst  s-Q>quit
```

The options view

The options view can be used to change the simulator's behavior. Different options are selected using the up and down cursor keys, and the options are changed using the left and right cursor keys.

Options include:

- Hardware

  As of this writing, the simulator only supports the "super 8051" option.

- Clock speed

  In addition to the several preset speeds, the clock speed can be set to a custom value by selecting the right-most option.

- Step mode

  The simulator can step a single cpu cycle (12 clocks for 8051) or one single instruction (1 to 4 cpu cycles)

- Exception enable

  All debug exceptions can be enabled or disabled separately.

Chapter 3. em8051 Internals

3.1. Source Organization

The em8051 project is split into two logical sections, the emulation core and the simulator front-end. It is possible to create new front-ends using just the emulation core.

The core consists of only three source files:

- core.c

  Main emulation source. Contains most of the public functions, and takes care of timer- and interrupt services, device resetting and intel hex file loading.

- disasm.c

  Contains handlers for disassembling opcodes.

- opcodes.c

  Contains handlers for emulating the opcodes.

The simulator front-end is split into several source files. The basic idea is that the main source file contains the global information and all the views have their own source files.

- emu.c

  Main front-end source. Contains most of the global data, misc. utility functions and the main loop.

- popups.c

  The source code of all of the different popup windows.

- mainview.c

  Main view related code and data.

- logicboard.c

  Logic board view related code and data.

- memeditor.c

  Memory editor view related code and data.

- options.c

  Options view related code and data. Also contains the global variables for options.

Additionally the emulation core and the simulator front end have one header file each for global information.

3.2. Emulation Core Notes

The emulation core was written to be as independent of the environment as possible. There are no memory allocations, and the interfaces are very simple. There is no global data and the source code is ANSI-C for maximum portability. Emulation accurancy was valued over emulation speed.

All of the data used by the emulator is stored in one structure. It would be trivial to simulate several 8051 cores at the same time due to this structure.

Instruction decoding is performed using a function pointer table, with one entry for each instruction. The 8051 has 256 different opcodes, but due to overlapping logic, only 111 functions needed to be written.

Optimization-wise the function pointer is a headache for branch prediction, and an alternate switch/case implementation was also written but it did not show much difference in performance based on profiling.

The 'reset' function builds the function pointer tables, so it has to be called before attempting to run 'tick'. Each call to 'tick' emulates one cpu cycle (i.e. 12 clock cycles on 8051).

3.3. Simulation Front-End Notes

Since the simulator only displays text, it was decided to make the simulator run in text mode. For this reason, the simulator uses ncurses, a standard text mode full-screen application library.

On windows, pdcurses was used instead, since there is no ncurses port for windows.

The ncurses library also makes it possible to run the simulator on a unix shell over ssh, or locally, or in mac os x console.

Since most of the front-end is UI code, it is a rather tangled mess. This is especially true for the main view, which simply has a lot going on. This could have been helped by writing the application in C++ instead, but that would have reduced the portability. The fact that ncurses code itself is rather cryptic does not help matters.

Everything except the emulation itself is implemented in the front-end code. This includes the run modes, breakpoints, and the additional widgets in the logic board view.