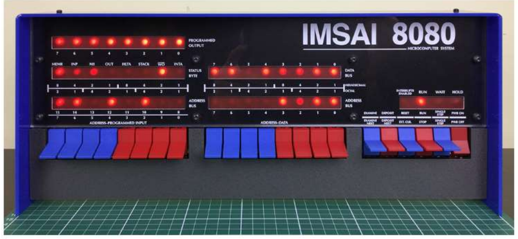
The emulation configuration switches, understanding and setting them.

This is a draft document. Please report errors, typos etc. on the forum (doc ver V6.1)

The IMSAI 8080 has two groups of 8 toggle switches which are used to enter addresses and data into the system. There is also a group of 6 switches used to control the system.



Showing the 3 groups of switches.

To change the behavior of the emulator we enter a special mode, set the two groups of 8 switches accordingly then save the configuration.

To change the configuration:

1. If you have any active work in the Web UI (such as editing a file under CP/M) close and save that work.
2. Turn off the IMSAI 8080 emulator (furthest ‘right’ switch to down) – all LEDs are now off.
3. Hold up the “EXAMINE” switch and keep holding it up while you…
4. Press and release the “Reset” button on the ESP32 module on the back of the system….
   a. This is not the “RESET” switch on the front of the system!
5. Keep holding “EXAMINE” up until the 4 LEDs on the right of the system start to cycle.
6. Release the “EXAMINE” switch, the current configuration is displayed on the LEDs above the switches.
7. Set the two banks of 8 switches to the pattern that matches your desired configuration.
8. Press the “DEPOSIT” switch up and release it.
9. The new configuration is displayed on the LEDs above the switches.
10. Press and release the “Reset” button on the ESP32 module on the back of the system.
    a. This is not the “RESET” switch on the front of the system!
11. Wait for the ESP32 to boot up, that can take a bit.
12. If needed connect to the system via your web browser.
13. Power up the IMSAI 8080 emulator, “PWR ON” switch is pressed up.
14. Test your new configuration.

Keep in mind that the LEDs indicate the current setting. Pressing the “DEPOSIT” switch up deposits the switch setting into the system. Setting the switches isn’t enough, you need to “DEPOSIT” the setting for the change to take effect.

Descriptions of the configuration switches. Note that Switch 15 is to the left of the machine.

Switch/Bit	Name	Description
15	Reserved	Unused – leave down/(0) – for future use
14	Reserved	Unused – leave down/(0) – for future use
13	Reserved	Unused – leave down/(0) – for future use
12	Reserved	Unused – leave down/(0) – for future use
11	NVS_BANK_ROM	Controls how the ROM is mapped into memory.
10	NVS_BOOT_ROM	Bits 10,9 and 8 set which of 7 ROMs is active
9	NVS_BOOT_ROM	Bits 10,9 and 8 set which of 7 ROMs is active
8	NVS_BOOT_ROM	Bits 10,9 and 8 set which of 7 ROMs is active
7	NVS_UNLIMITED	Controls emulated CPU speed (1)=Unlimited (0)=Use speed from bit 6
6	NVS_4MHZ	Set the CPU speed (0)=2MHz (1)=4MHz
5	NVS_NO_UNDOC	Allow emulation of undocumented CPU OpCodes (0)=Disable (1)=Enable
4	NVS_Z80	Select the emulated CPU type (0)=8080 (1)=Z-80
3	NVS_IF_STA	Wireless access type (0)=Access Point (1)=Station Node
2	NVS_LOG_LEVEL	Bits 2 and 1 select the error logging setting
1	NVS_LOG_LEVEL	Bits 2 and 1 select the error logging setting
0	NVS_POST	ESP32 “Power On Self Test” (0)=Disabled (1)=Enabled

Switches/Bits are 12,13,14 and 15 are reserved for future use and should be left Down/(0) at this time.

Switches/Bits 8,9,10 and 11 are the ROM Control bits and are used as a group.

If Switch/Bit 11 is down/(0) then the ROM is mapped to the ROMs starting address and MPU-B banked ROM/RAM is disabled.
If Switch/Bit 11 is up/(1) then the ROM is mapped at address 0000H and MPU-B banked ROM/RAM is enabled.

Switch/Bit 8,9 and 10 select which of the 7 available ROMs is active.
Switch/Bit 8,9,10 and 11 set to all down/(0) disables ROMs, all 64K of RAM is available.

To boot CPM the CPM 2.2 ROM is required. With the CPM 2.2 ROM (mpu-a-rom.hex) in slot ROM1 in the Web UI these switches/bits would be set as:

11	NVS_BANK_ROM	Up/(1) – CPM ROM at address 0000H
10	NVS_BOOT_ROM	Down/(0)
9	NVS_BOOT_ROM	Down/(0)
8	NVS_BOOT_ROM	Up/(1)

When the IMSAI 8080 is reset it starts by running the CPU instructions in the “CPM 2.2 ROM” which has been mapped into the CPUs address space by the settings above. The code in the ROM understands how to load the boot loader from the floppy disk in the A: drive which results in CP/M booting up.

To enable the 8K MBASIC 1.4 ROM. With the 8K MBASIC 1.4 in Slot 4 in the Web UI these switches/bits would be set as:

11	NVS_BANK_ROM	Down/(0) to enable the use of ROMs
10	NVS_BOOT_ROM	Up/(1)
9	NVS_BOOT_ROM	Down/(0)
8	NVS_BOOT_ROM	Down/(0)

When the IMSAI 8080 is reset it starts by running the CPU instructions in the “8K MBASIC 1.4” which has been mapped into the CPUs address space by the settings above. The code in the ROM is the MBASIC interpreter.

Switches/Bits 4,5,6,7 are the CPU control bits and are used as a group.

To set Z-80 CPU, 4MHz and enable undocumented OpCodes the switches/bits would be set as:

7	NVS_UNLIMITED	Down/(0)=Limited
6	NVS_4MHZ	Up/(1)=4MHz
5	NVS_NO_UNDOC	Up/(1)=Enable
4	NVS_Z80	Up/(1)=Z-80

To set 8080 CPU, 2MHz and disable undocumented OpCodes the switches/bits would be set as:

7	NVS_UNLIMITED	Down/(0)=Limited
6	NVS_4MHZ	Down/(0)=2MHz
5	NVS_NO_UNDOC	Down/(0)=Disable
4	NVS_Z80	Down/(0)=8080

A note about undocumented OpCodes.

The manufacturer of a microprocessor provides a list of valid OpCodes for the processor. These are the OpCodes the programmer would use when programming in assembly language. Both the 8080 and Z-80 are 8-bit machines. In 8 bits we have 256 unique bit patterns from 00000000 to 11111111. Each OpCode has a unique bit pattern however there is not a manufacture designated OpCode for every bit pattern. Through experimentation the undefined bit patterns can provide useful actions within the microprocessor. These then become the “Undocumented OpCodes”.

Switch/Bit 3 is used to control how the ESP32 interacts on your wireless network.

Down/(0) is the default AP mode. In this mode you need to discover the wireless network the ESP32 is broadcasting and connect to it. Opening the Web UI in this mode will only work once you are connected to the wireless network the ESP32 is broadcasting.
Once connected this way you can enter the SID and Password for your wireless network.

Up/(1) is the STA mode. The ESP32 will connect to the wireless network defined above and will become another device on your wireless network. You can connect directly to the system from your web browser without needing to join an additional wireless network as you need to do in AP mode.

Switches/Bits 1 and 2 control the logging level and are used as a group.

The combination of the 2 Switches/Bits sets the logging level.

Switch/Bit 0 controls the ESP POST settings.

Up/(1) will enable the “Power On Self-Test” to run on the ESP32.
Down/(0) disables the POST test.