Ho Ho Hoperating System (H3S) *v0.2020*

H3S End User Manual

This booklet is provided with the hope that it will be useful, but without any warranty because we don’t know how stupid you are. If you need support, read the manual. If you still need support after that, do an effort and look at it from a different angle. Rinse. Repeat.

Sincerely yours,
H3S Corp Customer Service

1. The H3S Computer

The H3S Computer has been built to last. To achieve that, we successfully downclocked our CPU to 50Hz. With a low amount of instruction per second, we reduce the energy consuption and your will to use the H3S Computer, which result in the computer lasting longer than what the customer really wants.

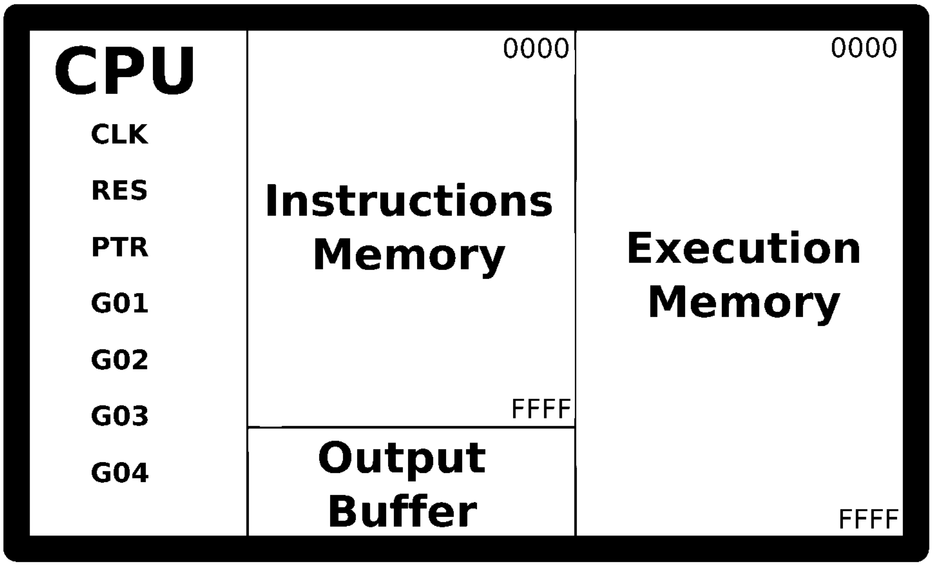
Please take good note that we do not accept return of H3S Computers that show signs of owner abuse.

2. The H3S Computer’s CPU

Our great CPU is a custom design optimized for testing the patience of our valued customers. We courteously included a schema of the CPU, a table with the usage of each register and another table with the usage of each memory region.

Register	Usage
CLK	Contains the number of clock cycle since the execution of the first instruction. Read-only.
RES	Contains the result of certain operations. Read-only.
PTR	Contains the address of the current instruction. Read-only.
Gxx	General-purpose register.

Memory region	Usage
Instruction memory IMEM	Contains the current program instructions and parameters. Addressing starts at 0x0000 and ends at 0xFFFF. 0x0000 must be a valid instruction and 0xFFFF should be an invalid instruction.
Execution memory MEM	Contains a copy of the current file. Addressing starts at 0x0000 and ends at 0xFFFF. See section 4 for more information about the file system.
Output buffer BUF	Contains bytes that will be printed on the next PRT. Append-only. See section 3 for more information about the instructions and the buffer system.



The initial value for each register is 0x0000, for instructions memory is the current program, for execution memory is 0x0000 and output buffer is considered empty.

Please note that every value is 16 bit in H3S.

3. Instructions for the H3S

The interact with the superior Hoperating System, you need a superior instruction set. Here it is, in all its glory :

Instruction	Parameters	Description
ADD	reg1 reg2	RES = reg2 + reg1
SUB	reg1 reg2	RES = reg2 - reg1
MUL	reg1 reg2	RES = reg2 * reg1
DIV	reg1 reg2	RES = reg2 / reg1
MOD	reg1 reg2	RES = reg2 % reg1
INC	reg	reg = reg + 1
DEC	reg	reg = reg - 1
SET	reg const	reg = const
GET	reg	Wait for a char from the user. Store in reg.
JMP	const	Jump to IMEM const.
JLT	const	Jump to IMEM const if RES < 0x0000
JGT	const	Jump to IMEM const if RES > 0x0000
JEQ	const	Jump to IMEM const if RES == 0x0000

AIB	reg	Append value in reg as an integer to BUF.
ACB	reg	Append value in reg as a char to BUF.
ASB	n/a	Append a space to BUF.
ALB	n/a	Append a new line to BUF.
PRT	n/a	Print BUF to screen and empty BUF.
SWP	reg1 reg2	Swap reg1 and reg2 values.
NUL	n/a	Do nothing for a cycle.
CPY	reg1 reg2	reg2 = reg1
AND	reg1 reg2	RES = reg2 & reg1
IOR	reg1 reg2	RES = reg2 | reg1
XOR	reg1 reg2	RES = reg2 ^ reg1
NOT	reg	RES = ~reg
CMP	reg1 reg2	Compare the values in reg1 and reg2. If reg1 > reg2 : RES = 0x0001 If reg1 == reg2 : RES = 0x0000 If reg1 < reg2 : RES = 0xFFFF
RTM	reg1 reg2	Copy value in reg2 to MEM[reg1].
MTR	reg1 reg2	Copy value in MEM[reg2] to reg1.
LOD	reg	Load file number reg to MEM. See section 4 for more information about the file system.
SAV	reg	Save MEM to file number reg. See section 4 for more information about the file system.

Please note that all code must be in capital letters.
Please note that reg, reg1 and reg2 should be replaced by the name of a register.
Please note that const should be a constant integer in hexadecimal.
Please note that a constant must be four character wide at all time.
Please note that an extra argument will be considered as a comment.

4. H3S’s file system

The H3S uses a particular file system to enable data persistance. There are a total of 65536 files numbered from 0000 to FFFF. Each file can store 65536 values. Files should never begin with 0x0000, to allow easy identification of empty files.

Please note that file 0000 will always be empty. It should be used for zeroing MEM when needed.