**HEC Model Z1 Rev 0.1.4Hardware Programming Manual**

# Pages

A page is a continuous region of memory. Each page is 256 bytes long. A total of 256 pages can exist in a single address space. Up to 128 independent address spaces are supported by the paging hardware. These pages are selected from 4096 pages or 1MiB of memory.

Two different pages within the address space can point to the same page on physical memory. In such case, and any operations performed on one page will be mirrored to the other page, since they are physically the same page.

Each page can be marked as Write-Protected (WP), Write-Interrupt (WI), Read-Interrupt (RI) and No-Opcode-Fetch (NX). Writes to a Write-Protected page is discarded and has no effect. The effects of other bits will be discussed in detail in the ‘Operating Modes’ section.

All pages contain RAM only, no MMIO is used in HEC, and no ROM is mapped to RAM space. ROM can be emulated by setting a page of RAM to write-protected. RAM is always mapped by the paging circuit. There is no way to turn off RAM mapping. A flat memory model can be emulated by setting each page entry in one address space to a different physical page.

# Operating Modes

## USR mode

Code running in user (USR) mode cannot perform IO operations. Running any IO instruction in user mode would have no effect and immediately trigger an NMI. NMI will also be triggered if the code tries to write to a Write-Interrupt page, or read from a Read-Interrupt page, or fetch an opcode from a No-Opcode-Fetch page. (caution: the Z80 is only consider the first cycle of an instruction an opcode fetch cycle, or M1 cycle. Therefore, later bytes in the instruction, such as address or immediate number, will not be considered as opcodes and will not trigger an NMI if read from a No-Opcode-Fetch page). A write to a write-protected will not generate an NMI unless the same page has been set to write-interrupt. External Interrupt can occur in USR mode as NMI.

User mode can be entered from system mode via either a read or a write access to the GOUSR port (port address: 0x07).

## SYS mode

Code running in system (SYS) mode can perform IO operations to access peripherals. Details of such operations will be discussed in detail in ‘Input and Output’ section.

All interrupts, internal or external, are masked in system mode. However, interrupts are still generated by the hardware. One case that should be paid special attention is the page attributes: interrupts will be generated as if the code is running in user mode. So, if page attribute bits are used to control access to the page (for example, using RI bit to prevent user programs from accessing sensitive information), system routines must clear internal interrupts after accessing those pages. This can be accomplished by either a read or a write to the CLRINT port (port address: 0x03).

System mode is entered via wither an RST instruction, or an NMI. Note that the HEC is not guaranteed to be in system mode immediately after reset. It is necessary that the programmer explicitly put the HEC into system mode during the POST process.

## ROM mode

The HEC is put into ROM mode immediately after a reset. In ROM mode, the lower half of the address space is 32KiB of onboard boot ROM, and the upper half is empty. Technically ROM mode and USR/SYS modes are completely orthogonal, however, there is little to no point in putting the HEC into user mode before exiting ROM mode.

The onboard ROM is not mapped by the paging circuit and cannot be written to. Any write in ROM mode will be directed to RAM and is subject to RAM mapping and attribute bits. Therefore, it is necessary to initialize the paging circuit before writing to RAM.

In ROM mode, all read access to external IO devices (expansion cards) will be redirected to ROM read. The high byte of the address bus is used as lower address byte to ROM, and 5 IO port bits are used as higher address bits of ROM. This setup facilitates the use of a INDR instruction to quickly copy ROM on a card into RAM.