

README on the IOBARRIER for CardEngine IO

Due to an unfortunate oversight when the Card Engines were designed, the signals that control access to some peripherals, most notably the SMC91C9111 ethernet controller, are not properly handled.

The symptom is that some back to back IO with the peripheral returns unreliable data. With the SMC chip, you'll see errors about the bank register being 'screwed'.

The cause is that the AEN signal to the SMC chip does not transition for every memory access. It is driven through the CPLD from the CS7 line of the CPU's static memory controller which is optimized to eliminate unnecessary transitions. Yet, the SMC requires a transition for every write access. The Sharp website has more information about the effect this power-conserving feature has on peripheral interfacing.

The solution is to follow every write access to the SMC chip with an access to another memory region that will force the CPU to release the chip select line. It is important to guarantee that this access forces the CPU off-chip. We map a page of SDRAM as if it were an uncacheable IO device and read from it after every SMC IO write operation.

SMC IO
BARRIER IO

Only this sequence is important. It does not matter that there is no BARRIER IO before the access to the SMC chip because the AEN latch only needs occurs after the SMC IO write cycle. The routines that implement this work-around make an additional concession which is to disable interrupts during the IO sequence. Other hardware devices (the LogicPD CPLD) have registers in the same physical memory region as the SMC chip. An interrupt might allow an access to one of those registers while SMC IO is being performed.

You might be tempted to think that we have to access another device attached to the static memory controller, but the empirical evidence indicates that this is not so. Mapping 0x00000000 (flash) and 0xc0000000 (SDRAM) appear to have the same effect. Using SDRAM seems to be faster. Choosing to access an undecoded memory region is not desirable as there is no way to know how that chip select will be used in the future.