

I/O addressing

Once your peripheral-card program has the slot number, the card can use the number to address the I/O locations allocated to the slot. Table 6-4 shows how these locations are related to 16 base addresses starting with \$C080. Notice that the difference between the base address and the desired I/O location has the form \$n0, where n is the slot number. Starting with the slot number in the accumulator, the following example computes this difference by four left shifts, then loads it into an index register and uses the base address to specify one of 16 I/O locations.

```
ASL                ; get n into
ASL                ;
ASL                ;
ASL                ; ...high-order nybble
TAX                ; ...of index register
LDA    $C080,X    ; load from first I/O location
```

❖ *Selecting your target:* You must make sure that you get an appropriate value into the index register when you address I/O locations this way. For example, starting with 1 in the accumulator, the instructions in the above example perform an LDA from location \$C090, the first I/O location allocated to slot 1. If the value in the accumulator had been 0, the LDA would have accessed location \$C080, thereby setting the soft switch that selects the second bank of RAM at location \$D000 and enables it for reading.

See the section "Setting Bank Switches" in Chapter 4 for more information.

Table 6-4
Peripheral-card I/O base addresses

Base address	Connector number						
	1	2	3	4	5	6	7
\$C080	\$C090	\$C0A0	\$C0B0	\$C0C0	\$C0D0	\$C0E0	\$C0F0
\$C081	\$C091	\$C0A1	\$C0B1	\$C0C1	\$C0D1	\$C0E1	\$C0F1
\$C082	\$C092	\$C0A2	\$C0B2	\$C0C2	\$C0D2	\$C0E2	\$C0F2
\$C083	\$C093	\$C0A3	\$C0B3	\$C0C3	\$C0D3	\$C0E3	\$C0F3
\$C084	\$C094	\$C0A4	\$C0B4	\$C0C4	\$C0D4	\$C0E4	\$C0F4
\$C085	\$C095	\$C0A5	\$C0B5	\$C0C5	\$C0D5	\$C0E5	\$C0F5
\$C086	\$C096	\$C0A6	\$C0B6	\$C0C6	\$C0D6	\$C0E6	\$C0F6
\$C087	\$C097	\$C0A7	\$C0B7	\$C0C7	\$C0D7	\$C0E7	\$C0F7
\$C088	\$C098	\$C0A8	\$C0B8	\$C0C8	\$C0D8	\$C0E8	\$C0F8
\$C089	\$C099	\$C0A9	\$C0B9	\$C0C9	\$C0D9	\$C0E9	\$C0F9