680x0: How It Transfers a 32-bit Operand Over a 16-bit Data Path

Article Created: 23 December 1992

TOPIC ---

How does a Motorola microprocessor transfer a 32-bit operand over a 16-bit data path?

DISCUSSION ---

The 68020 and its successors implement a full 32-bit architecture. The 680x0 has implemented 32-bit registers along with a 32-bit data and address buses. The processor supports a dynamic bus sizing mechanism that allows the processor to transfer operands to or from external devices while automatically determining the device path size on a cycle-by-cycle basis. The dynamic bus interface allows for a simple highly efficient access to devices of differing data path widths. In dealing with transfers, a "path" refers to the external data bus width at the slave device (memory, peripheral, etc.).

You can think of it this way: the 680x0 gives you enough room to build a four-lane highway (path) for the address bus and the data bus. It is up to the manufacturer (Apple) to decide how many lanes to build for each path. The cost (engineering time, logic board denisty, component cost, time to market, etc.) of the path goes up with each lane added. For example, Apple may decide to build a system that supports 32-bit address path and a 16-bit data path.

The processor allows operand transfers to or from 8-, 16-, and 32-bit paths by dynamically determining the path size during each bus cycle. For example, if the processor is executing an instruction that requires a 32-bit operand, it attempts to read 32 bits during the first bus cycle. If the path responds that it is 16 bits wide, the processor latches the 16 bits of valid data and runs another cycle to obtain the other 16 bits. An 8-bit path is handled similarly, but with four read cycles.

Keywords: <None>

==========

19960215 11:05:19.00

Tech Info Library Article Number: 11135