ADDRESS MULTIPLY CONCEPTS

The CRAY-2 Address Multiply functional unit is located on the AM module. The multiply unit performs 32-bit integer multiplication of two A register operands.

The Address Registers, which are both the source and destination for the multiply unit, are located on the AR module. The operands are received by the multiply unit in two sequential clock periods, since there is only a single path from the registers to the unit. Ak is sent to the multiply unit first followed by Aj in the next clock period. The destination register is Ai.

The multiplication of two 32-bit operands could yield up to a 65-bit result. The result register is a 32-bit A register. The result that is sent to the A register is the lower 32-bits of the result of the multiplicaton process. Any higher order bits are ignored and there is no indication of an error. The multiply pyramid is therefore only 32-bits wide and includes the bits that make up the lower 32-bits of the multiplication result.

In order to save a considerable amount of logic in the address multiply unit, the 32-bits of Ak data is divided into a group of even data bits and a group of odd data bits. The even bits of Ak data are processed with all bits of Aj data first. The odd bits of Ak data are processed with all bits of Aj data second. The processing of odd Ak data lags the processing of even Ak data by one clock period. The two groups of processed data are merged in the later ranks of the multiply unit.

The multiply unit receives only one control signal. Go Multiply is the control signal from the JA module. This signal is used to divide the Ak operand into odd and even data at the proper time. Go Multiply is also used to hold the Aj data in the input register, for one clock, to allow for the two sequential clock periods of data processing. This control signal is also delayed and used to gate the multiply result off to the AR module at the proper time.

The gating of the multiply result through the result output latches is necessary since the path from the AM module to the AR is shared with P register data. The AM module receives a 32-bit P register value from the IB module. This value is passed on to the AR module, over the common path, and is written to Ai as a return jump address for an 002 instruction. The AM module also sends a copy of the P register value to the EA module, after a one clock stop over. This value can be latched up on the EA and sent either to Common Memory or to the Foreground Processor by executing specific foreground functions.