Dr W.Y Stevens

April 24, 1961

The 8108 High Speed Floating Point Attachment

1. General Description

The 8108 attachment to the 8106 machine provides a completely program-compatible means of performing high speed floating point arithmetic. The following table indicates the increase in performance obtained by attaching the 8108 to the 8106 CPU.

Instruction	8106 Instruction Time		8108 Instruction Time	
	8 - digit precision	12 - digit precision	8 - digit precision	12-digit precision
RESET	26	34	10	10
ADD	26	34	10	10
MULTIPLY	100	280	19	24
DIVIDE	274	880	24	28
Indexing	7	7	3	3

The 8108 is strictly a parallel floating point engine, equipped with a double length (24 hexadecimal digit) accumulator and facilities for sign and exponent manipulation. All decoding of instructions, address modification other than indexing, and transferring of data between the 8108 and memory remain the responsibility of the 8106, and the times for such functions remain unaffected.

2. Relation of the 8108 Accumulator to Location P+4

Since the 8108 has a hardware accumulator and the 8106 has an implied accumulator at memory location P+4, certain conditions must be specified under which the 8108 accumulator is stored at location P+4, and vice versa, P+4 is loaded into the hardware accumulator, if strict program compatibility is to be maintained. To this end a trigger is provided which is off or on accordingly as the accumulator contents are in P+4 or in 8108 hardware. The situations resulting in the shuttling of the accumulator contents between the 8108 accumulator register and storage location P+4 are described below.

a. The 8108 accumulator is stored in P+4 and the trigger is turned off when

the trigger is on and

(i) the current instruction is multiple address

or

(ii) the first effective address for the current instruction is P+4

or

(iii) there is an interruption.

b. Location P+4 is loaded into the 8108 accumulator and the trigger is turned on when

the trigger is off and

the current instruction is a floating point or store accumulator instruction.

c. The load accumulator instruction loads the 8108 accumulator and turns the trigger on.

The effect of the above arrangements is to leave the accumulator contents in the 8108 accumulator registers for strings of floating point instructions, and to leave the 8108 accumulator register dormant for strings of non-floating point instructions. The 8108 accumulator register is stored in location P+4 for any instruction, including a floating point instruction whose first effective address is P+4. This, for example, preserves the legitimacy of squaring the contents of the accumulator by means of a floating multiply instruction addressing location P+4.

3. Double Precision Arithmetic

In order to take advantage of the double precision potential inherent in the 8108 double length accumulator, a double precision mode of operation indicated in the PCW is provided. There are no extra instruction codes, but the floating point instructions are interpreted slightly differently in double precision mode. Factors common to all the floating point instructions in double precision mode are:

(i) Whenever the high order accumulator fraction and accumulator exponent are stored in P+4, the low order accumulator fraction will be stored in location P+5. Similarly, whenever location P+4 is loaded into the 8108 accumulator, the fractional field of location P+5 will be loaded into the low order accumulator. The exact conditions under which this storing and loading take place have been described above.

(ii) During normalization of a floating-point result, the entire 24 digit accumulator fraction is shifted. Only if the entire result fraction is zero will the Generated Extremum Negative indicator be turned on, and the extremum bit be given the value one.

There is no double precision noisy mode and no 8-digit double precision mode.

Comments pertinent to the individual floating point instructions and the LOAD/STORE ACCUMULATOR instructions are listed below:

LOAD ACCUMULATOR

This instruction will load the high order accumulator and clear the low order accumulator.

STORE ACCUMULATOR

This instruction will store the high order accumulator and leave the low order accumulator undisturbed.

RESET ADD

The low order accumulator is cleared at the start of this instruction.

ADD

If the exponent of the operand from storage is greater than or equal to the exponent of the operand in the accumulator, the double precision accumulator operand will be added to the single precision storage operand to give a double precision sum. If however, the storage exponent is less than the accumulator exponent, the low order fraction of the accumulator operand is lost. In this case, a single precision accumulator operand is added to a single precision storage operand to give a double precision sum.

MULTIPLY

The low order fraction of the accumulator operand is ignored. A single precision accumulator operand is multiplied by a single precision storage operand to yield a double precision product.

DIVIDE

A double precision dividend in the accumulator is divided by a single precision divisor from storage to yield a single precision quotient. The remainder is not available. It may be obtained by programming. At the completion of a division, the low order accumulator fraction is cleared.

The direct or indirect addressing, indexing, sign, and normalized or unnormalized modifications to the basic floating point instructions are all available in the double precision mode.

No special load and store instructions are provided for the low order accumulator. The low order accumulator fraction is loaded and stored by programming. For example, if both the high and low order fractions of a double precision number are supplied with exponents, then the complete double precision number may be loaded into the accumulator by a Load Accumulator instruction followed by an Add instruction. The low order fraction may be moved into the high order accumulator position by first storing the high order part and then subtracting it from the accumulator, and normalizing the result.

With the double precision mode operations as described, double precision floating point operations, in which double precision results are obtained from double precision operands, may conveniently be programmed.

D. S. Henderson

D. S. Henderson
Systems Planning

DSH/pkb