Information on How to Calculate "Composite Theoretical Performance ("CTP")

Technical Note:

"COMPOSITE THEORETICAL PERFORMANCE" ("CTP")

Abbreviations used in this Technical Note

"CE"	"computing element"(typically an arithmetic logical unit)
FP	floating point
XP	fixed point
t	execution time
XOR	exclusive OR
CPU	central processing unit
TP	theoretical performance (of a single "CE")
"CTP"	"composite theoretical performance" (multiple "CEs")
R	effective calculating rate
WL	word length
L	word length adjustment
*	multiply

Execution time t is expressed in microseconds, TP and "CTP" are expressed in millions of theoretical operations per second (Mtops) and WL is expressed in bits.

Outline of "CTP" calculation method

"CTP" is a measure of computational performance given in Mtops. In calculating the "CTP" of an aggregation of "CEs" the following three steps are required:

1. Calculate the effective calculating rate R for each "CE";

2. Apply the word length adjustment (L) to the effective calculating rate (R), resulting in a Theoretical Performance (TP) for each "CE";

3. If there is more than one "CE", combine the TPs, resulting in a "CTP" for the aggregation.

Details for these steps are given in the following sections.

Note 1: For aggregations of multiple "CEs" that have both shared and unshared memory subsystems, the calculation of "CTP" is completed hierarchically, in two steps: first, aggregate the groups of "CEs" sharing memory; second, calculate the "CTP" of the groups using the calculation method for multiple "CEs" not sharing memory.

Note 2: "CEs" that are limited to input/output and peripheral functions (e.g., disk drive, communication and video display controllers) are not aggregated into the "CTP" calculation.

The following table shows the method of calculating the Effective Calculating Rate R for each "CE":

Step 1: *The effective calculating rate R*

For "CEs" Implementing: **Note:** Every "CE" must be evaluated independently.	Effective calculating Rate, R
XP only (R_{xp})	1 --- $3 * (t_{xp\ add})$ if no add is implemented use: 1 --- $(t_{xp\ mult})$ If neither add nor multiply is implemented use the fastest available arithmetic operation as follows: 1 --- $3 * t_{xp}$ See Notes X & Z
FP only (R_{fp})	$\max\left(\frac{1}{t_{fp\ add}}, \frac{1}{t_{fp\ mult}}\right)$ See Notes X & Y
Both FP and XP (R)	Calculate both R_{xp}, R_{fp}
For simple logic processors not implementing any of the specified arithmetic operations.	1 --- $3 * t_{log}$ Where t_{log} is the execute time of the XOR, or for logic hardware not implementing the XOR, the fastest simple logic operation. See Notes X & Z
For special logic processors not using any of the specified arithmetic or logic operations.	R = R' * WL/64 Where R' is the number of results per second, WL is the number of *bits* upon which the logic operation occurs, and 64 is a factor to normalize to a 64 bit operation.

Note W: For a pipelined "CE" capable of executing up to one arithmetic or logic operation every clock cycle after the pipeline is full, a pipelined rate can be established. The effective calculating rate (R) for such a "CE" is the faster of the pipelined rate or non-pipelined execution rate.

Note X: For a "CE" that performs multiple operations of a specific type in a single cycle (e.g., two additions per cycle or two identical logic operations per cycle), the execution time t is given by:

$$t = \frac{\text{cycle time}}{\text{the number of identical operations per machine cycle}}$$

"CEs" that perform different types of arithmetic or logic operations in a single machine cycle are to be treated as multiple separate "CEs" performing simultaneously (e.g., a "CE" performing an addition and a multiplication in one cycle is to be treated as two "CEs", the first performing an addition in one cycle and the second performing a multiplication in one cycle). If a single "CE" has both scalar function and vector function, use the shorter execution time value.

Note Y: For the "CE" that does not implement FP add or FP multiply, but that performs FP divide:

$$R_{fp} = \frac{1}{t_{fpdivide}}$$

If the "CE" implements FP reciprocal but not FP add, FP multiply or FP divide, then

$$R_{fp} = \frac{1}{t_{fpreciprocal}}$$

If none of the specified instructions is implemented, the effective FP rate is 0.

Note Z: In simple logic operations, a single instruction performs a single logic manipulation of no more than two operands of given lengths. In complex logic operations, a single instruction performs multiple logic manipulations to produce one or more results from two or more operands.

Rates should be calculated for all supported operand lengths considering both pipelined operations (if supported), and non-pipelined operations using the fastest executing instruction for each operand length based on:

1. Pipelined or register-to-register operations. Exclude extraordinarily short execution times generated for operations on a predetermined operand or operands (for example, multiplication by 0 or 1). If no register-to-register operations are implemented, continue with (2).

2. The faster of register-to-memory or memory-to-register operations; if these also do not exist, then continue with (3).

3. Memory-to-memory.

In each case above, use the shortest execution time certified by the manufacturer.

Step 2: *TP for each supported operand length WL*

Adjust the effective rate R (or R') by the word length adjustment L as follows:

TP = R * L, where L = (1/3 + WL/96)

Note: The word length WL used in these calculations is the operand length in bits. (If an operation uses operands of different lengths, select the largest word length.) The combination of a mantissa ALU and an exponent ALU of a floating point processor or unit is considered to be one "CE" with a Word Length (WL) equal to the number of bits in the data representation (typically 32 or 64) for purposes of the "CTP" calculation.

This adjustment is not applied to specialized logic processors that do not use XOR instructions. In this case TP = R.

Select the maximum resulting value of TP for:

Each XP-only "CE" (R_{xp});

Each FP-only "CE" (R_{fp});

Each combined FP and XP "CE" (R);

Each simple logic processor not implementing any of the specified arithmetic operations; *and*

Each special logic processor not using any of the specified arithmetic or logic operations.

Step 3: *"CTP" for aggregations of "CEs", including CPUs.*

For a CPU with a single "CE", "CTP" = TP
(for "CEs" performing both fixed and floating point operations
TP = max (TP_{fp}, TP_{xp}))

"CTP" for aggregations of multiple "CEs" operating simultaneously is calculated as follows:

Note 1: For aggregations that do not allow all of the "CEs" to run simultaneously, the possible combination of "CEs" that provides the largest "CTP" should be used. The TP of each contributing "CE" is to be calculated at its maximum value theoretically possible before the "CTP" of the combination is derived.

N.B.: To determine the possible combinations of simultaneously operating "CEs", generate an instruction sequence that initiates operations in multiple "CEs", beginning with the slowest "CE" (the one needing the largest number of cycles to complete its $C_2 = C_3 = C_4 = ... = C_n = 0.75$

Note 1: When the "CTP" calculated by the above method does not exceed 194 Mtops, the following formula may be used to calculate C_i:

operation) and ending with the fastest "CE". At each cycle of the sequence, the combination of "CEs" that are in operation during that cycle is a possible combination. The instruction sequence must take into account all hardware and/or architectural constraints on overlapping operations.

Note 2: A single integrated circuit chip or board assembly may contain multiple "CEs".

Note 3: Simultaneous operations are assumed to exist when the computer manufacturer claims concurrent, parallel or simultaneous operation or execution in a manual or brochure for the computer.

Note 4: "CTP" values are not to be aggregated for "CE" combinations (inter)connected by "Local Area Networks", Wide Area Networks, I/O shared connections/devices, I/O controllers and any communication interconnection implemented by "software".

Note 5: "CTP" values must be aggregated for multiple "CEs" specially designed to enhance performance by aggregation, operating simultaneously and sharing memory,- or multiple memory/"CE"- combinations operating simultaneously utilizing specially designed hardware.

This aggregation does not apply to "electronic assemblies" described by 4A003.c.

"CTP" = $TP_1 + C_2 * TP_2 + ... + C_n * TP_n$,

where the TPs are ordered by value, with TP_1 being the highest, TP_2 being the second highest, ..., and TP_n being the lowest. C_i is a coefficient determined by the strength of the interconnection between "CEs", as follows:

For multiple "CEs" operating simultaneously and sharing memory:

$$C_i = \frac{0.75}{\sqrt{m}} \qquad (i = 2, ..., n)$$

where m = the number of "CEs" or groups of "CEs" sharing access.

provided:

1. The TP_1 of each "CE" or group of "CEs" does not exceed 30 Mtops;

2. The "CEs" or groups of "CEs" share access to main memory (excluding cache memory) over a single channel; *and*

3. Only one "CE" or group of "CEs" can have use of the channel at any given time.

N.B.: This does not apply to items controlled under Category 3.

Note 2: "CEs" share memory if they access a common segment of solid state memory. This memory may include cache memory, main memory or other internal memory. Peripheral memory devices such as disk drives, tape drives or RAM disks are not included.

For Multiple "CEs" or groups of "CEs" not sharing memory, interconnected by one or more data channels:

$C_i = 0.75 * k_i$ (i = 2, ... , 32) (see Note below)
$= 0.60 * k_i$ (i = 33, ... , 64)
$= 0.45 * k_i$ (i = 65, ... , 256)
$= 0.30 * k_i$ (i > 256)

The value of C_i is based on the number of "CE"s, not the number of nodes.

where k_i = min (S_i/K_r, 1), and
K_r = normalizing factor of 20 MByte/s.
S_i = sum of the maximum data rates (in units of MByte/s) for all data channels connected to the i^{th} "CE" or group of "CEs" sharing memory.

When calculating a C_i for a group of "CEs", the number of the first "CE" in a group determines the proper limit for C_i. For example, in an aggregation of groups consisting of 3 "CEs" each, the 22nd group will contain "CE"$_{64}$, "CE"$_{65}$ and "CE"$_{66}$. The proper limit for C_i for this group is 0.60.

Aggregation (of "CEs" or groups of "CEs") should be from the fastest-to-slowest; i.e.:

$TP_1 \geq TP_2 \geq .... \geq TP_n$, and

in the case of $TP_i = TP_{i+1}$, from the largest to smallest; i.e.:

$C_i \geq C_{i+1}$

Note: The k_i factor is not to be applied to "CEs" 2 to 12 if the TP_i of the "CE" or group of "CEs" is more than 50 Mtops; i.e., C_i for "CEs" 2 to 12 is 0.75.