

Technical Note on "CTP":

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})	$\frac{1}{3 \times (t_{xp \text{ add}})}$ if no add is implemented use: $\frac{1}{(t_{xp \text{ mult}})}$ if neither add nor multiply is implemented use the fastest available arithmetic operation as follows: $\frac{1}{3 \times t_{xp}}$ See Notes X & Z
FP only (R _{fp})	Max $\frac{1}{t_{fp \text{ add}}}, \frac{1}{t_{fp \text{ mult}}}$ 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.	$\frac{1}{3 \times t_{\text{log}}}$ 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" which 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" which 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:

$$R_{fp} = \frac{1}{t_{fp \text{ divide}}}$$

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

$$R_{fp} = \frac{1}{t_{fp \text{ reciprocal}}}$$

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.