Proposal for IEEE Standard Full-Spatial Floating-Point Arithmetic.

Drafted on Tuesday, January 28, 2020 07:00-MT
Drafted by Dr. Thomas Catalano
<tomcatalano.0@gmail.com>

Proposal for IEEE Std n-2020 for Full-Spatial Floating-Point Arithmetic is a working draft.
ieee?-2020-01.0x(system structure for 1to1) for a working standard.

This standard proposes formats and operations for full-spatial floating-point arithmetic in the newly designed full-spatial(median-free liquid via photonic bit transfer) processors/computer systems.

Diversity: As we continued into 21st century yearning for seamless interconnectivity and quasi-operational machines new designs were needed. One such designed engineering utility was invented by Dr. Thomas Catalano, the Quantum Computer Full-Spatial median-free liquid photonic bit transfer computational processor/computer so to the guidelines to coherently run arbitrary instruction sets.

Standards: With the new Full-Spatial computational processors/computers a conjoined effort to efficiently commence usage processing.
A core processing unit runs single and/or overlayed data stream(s) as RGB color spectrum number sequencing as one "channeled" stream like a thread in a microprocessor. The "channeled" stream is Drsequencing data numerically by referencing the RGB color spectrum.
In waiting on a timeline for us converging to materialize a working group I will continue development of possible arithmetic structures for architecture based on design and engineering;
Observe...
//

/
Need new requirements to define unit as per independent of 360degree area. No matter the size of cavity, defining measurable rate is compared by data rate (d rate) to "defined" transmitting median(tm rate).
/

/

-requires unit of measurement for RGB color/spectrum(by)data stream coverage to compute rate() per time/frame.(ie a spectra unit).

A required unit of measurement for one(1) spectra unit as per IEEE would define the scale to calculate 0-100% cofactor..

()3×256=768×360=276480 by one(n area) cavity or multi(n area) cavities spectrums =

data at()
×
per second of ON time=

/

=

([(Floating-Point*/0×
Floating-Point*/0)])

=

((3×256)[x*])(k*))]

/
new programming language required for pragmatic system requirements needed for operations/conversions to...
Now;

-we fill in with architecture hardware specifications..
[]
[]
[]
[]
[]
[]
[]
[]
[]

/
All (spectral areas(undetermined defined range) by(×)×(the cavity(x)) are deterministic[x] and accountable (per unit(xOFn)) of(per unit filled(xOFn)=(spectral range(3×256)×(n))...
/

Defined area of spectral area as standard;
-material per unit sq. ?...
/
then-
/
-we know we can not use the ranges defined by present standards or programming languages required to define the future approved ranges of grouped numbers then to be translated into current microprocessor architecture.
/
/
Example- progmatic inference;
/
Assembler's to (say...i dont know.?. "Qued" lang for the Full-Spatial Quantum Computer);

- Due to new architecture non-restrictions setting implicit charters are required…

1. Looped verification of spectral timed events
2. noBounds vs. nLatency
3. Numbered/naming power of (n) number sequensing in blocks named(x)

/
Example-
A future example conversion of current assemblers.
/
GNU Assembler
/
.gprel32 expression-
Computes the difference between the address in expression and the GP for the current object file, and stores it in 4 bytes. In addition to being smaller than a full 8 byte address, this also does not require a dynamic relocation when used in a shared library.
-
.t_floating expression
Stores expression as an ieee double precision value.
-
.s_floating expression
Stores expression as an ieee single precision value.
-
mfpu=floating-point-format
This option specifies the floating point format to assemble for. The assembler will issue an error message if an attempt is made to assemble an instruction which will not execute on the target floating point unit. The following

format options are recognized: softfpa, fpe, fpe2, fpe3, fpa, fpa10, fpa11,arm7500fe, softvfp, softvfp+vfp, vfp, vfp10, vfp10-r0, vfp9, vfpxd,
vfpv2, vfpv3, vfpv3-d16, vfpv4, vfpv4-d16, fpv4-sp-d16, arm1020t,
arm1020e, arm1136jf-s, maverick, neon, and neon-vfpv4.
In addition to determining which instructions are assembled, this option also affects the way in which the .double assembler directive behaves when assembling little-endian code.
The default is dependent on the processor selected. For Architecture 5 or later, the default is to assembler for VFP instructions; for earlier architectures the default is to assemble for FPA instructions.
/
9.3.3 Floating Point
The ARM family uses ieee floating-point numbers.
9.3.3.1 ARM relocation generation
Specific data relocations can be generated by putting the relocation name in parentheses after the symbol name. For example:
.word foo(TARGET1)
This will generate an ‘R_ARM_TARGET1’ relocation against the symbol foo. The following relocations are supported: GOT, GOTOFF, TARGET1, TARGET2, SBREL, TLSGD, TLSLDM, TLSLDO,
GOTTPOFF and TPOFF.
For compatibility with older toolchains the assembler also accepts (PLT) after branch targets. This will generate the deprecated ‘R_ARM_PLT32’ relocation.
Relocations for ‘MOVW’ and ‘MOVT’ instructions can be generated by prefixing the value with ‘#:lower16:’ and ‘#:upper16’ respectively. For example to load the 32-bit address of foo into r0:
MOVW r0, #:lower16:foo
MOVT r0, #:upper16:foo
9.3.4 ARM Machine Directives
.2byte expression [, expression]*
.4byte expression [, expression]*
.8byte expression [, expression]*
These directives write 2, 4 or 8 byte values to the output section.
.align expression [, expression]
This is the generic .align directive. For the ARM however if the first argument is zero (ie no alignment is needed) the assembler will behave as if the argument had been 2 (ie. pad to the next four byte boundary). This is for compatibility
with ARM’s own assembler
/
.arch name
/
name .qn register name [.type] [[index]]
The dn and qn directives are used to create typed and/or indexed register aliases

for use in Advanced SIMD Extension (Neon) instructions. The former should be used to create aliases of double-precision registers, and the latter to create aliases of quad-precision registers.
If these directives are used to create typed aliases, those aliases can be used in Neon instructions instead of writing types after the mnemonic or after each operand. For example:
x .dn d2.f32
y .dn d3.f32
z .dn d4.f32[1]
vmul x,y,z
This is equivalent to writing the following:
vmul.f32 d2,d3,d4[1]
Aliases created using dn or qn can be destroyed using unreq.
.eabi_attribute tag, value
Set the EABI object attribute tag to value.
The tag is either an attribute number, or one of the following: Tag_CPU_raw_name, Tag_CPU_name, Tag_CPU_arch, Tag_CPU_arch_profile, Tag_ARM_ISA_use, Tag_THUMB_ISA_use, Tag_FP_arch, Tag_WMMX_arch, Tag_Advanced_SIMD_arch, Tag_PCS_config, Tag_ABI_PCS_R9_use, Tag_ABI_PCS_RW_data, Tag_ABI_PCS_RO_data, Tag_ABI_PCS_GOT_use, Tag_ABI_PCS_wchar_t, Tag_ABI_FP_rounding, Tag_ABI_FP_denormal, Tag_ABI_FP_exceptions, Tag_ABI_FP_user_exceptions, Tag_ABI_FP_number_model, Tag_ABI_align_needed, Tag_ABI_align_preserved, Tag_ABI_enum_size, Tag_ABI_HardFP_use, Tag_ABI_VFP_args, Tag_ABI_WMMX_args, Tag_ABI_optimization_goals, Tag_ABI_FP_optimization_goals, Tag_compatibility, Tag_CPU_unaligned_access, Tag_FP_HP_extension, Tag_ABI_FP_16bit_format, Tag_MPextension_use, Tag_DIV_use, Tag_nodefaults, Tag_also_compatible_with, Tag_conformance, Tag_T2EE_use, Tag_Virtualization_use
The value is either a number, "string", or number, "string" depending on the tag.
.even This directive aligns to an even-numbered address.
.extend expression [, expression]*
.ldouble expression [, expression]*
These directives write 12byte long double floating-point values to the output section. These are not compatible with current ARM processors or ABIs.
/
Example-

GNU Assembler FSC(Full-Spatial Compute).

1. T
2. X
3. R

/
/
/
Formatting/Transfer from the "Qued" lang programming language to multiple architecture-

-Registry requirements per area of available spectral cohesion rate..

-library's to systematically bind sums from proposed IEEE Full-Spatial Floating-Point Arithmetic to file format and/or runing sums to target architecture whether dependent to specific output of Full-Spatial processor/computer(ie operating system specific to FSPC) or available architecture..
/
/
/
//