On the Flop and Flap Counts of the 2,8-Split-Radix FFT

Paul Caprioli

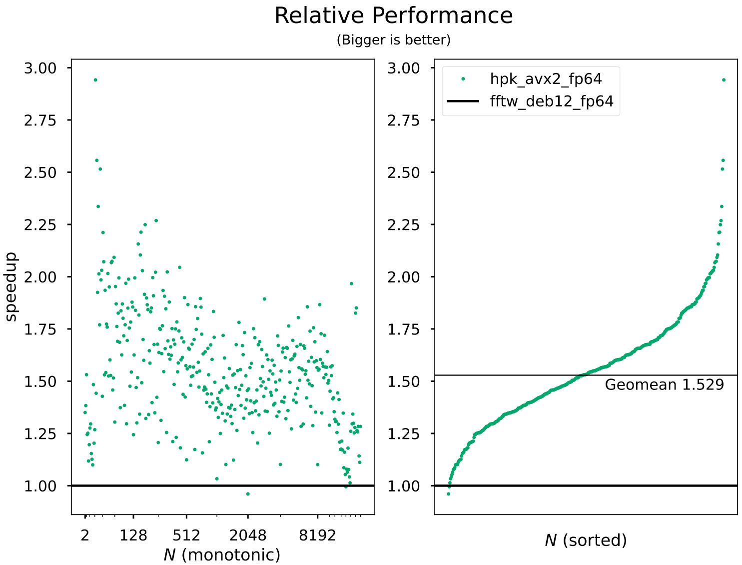
https://hpkfft.com

2025 May 5
in the West Texas town of El Paso

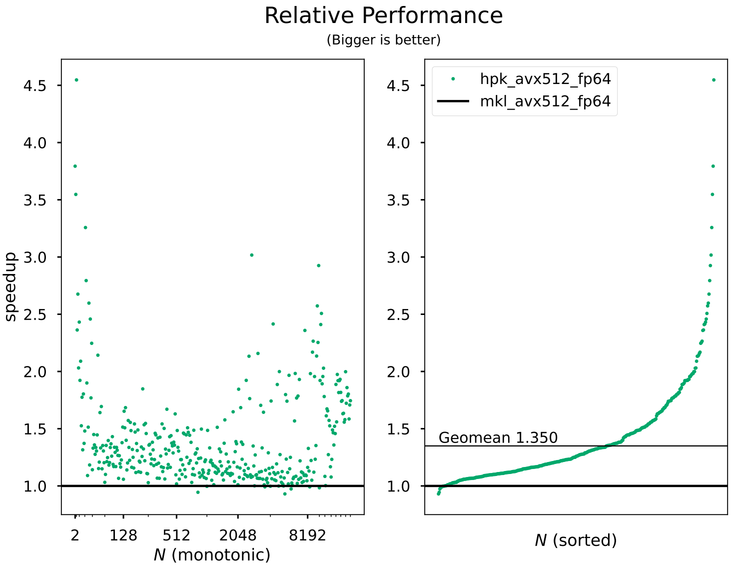
“Running Gun”

I knew someday I’d meet him,
    for his hand like lightning flashed.
My own gun stood in leather,
    as his bullet tore its path.
As my strength was slowly fading,
    I could see him walk away.
And I knew that where I lie tonight,
    he too must lie some day.

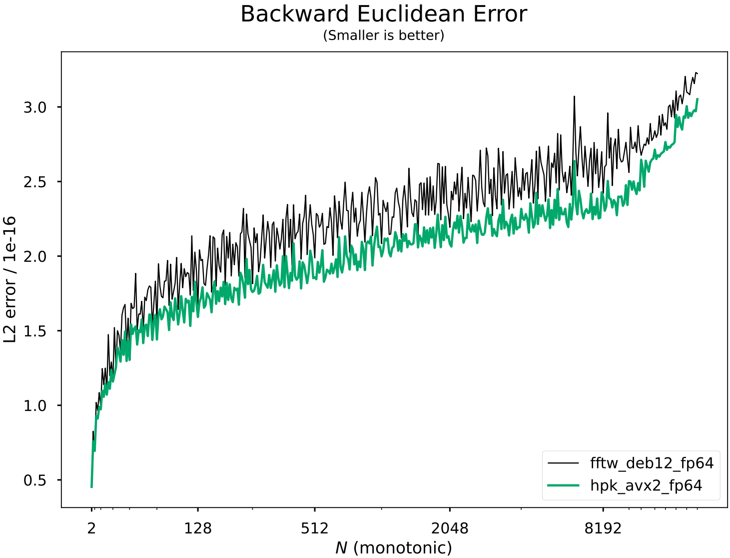
Performance: AVX2 Double Precision vs. FFTW



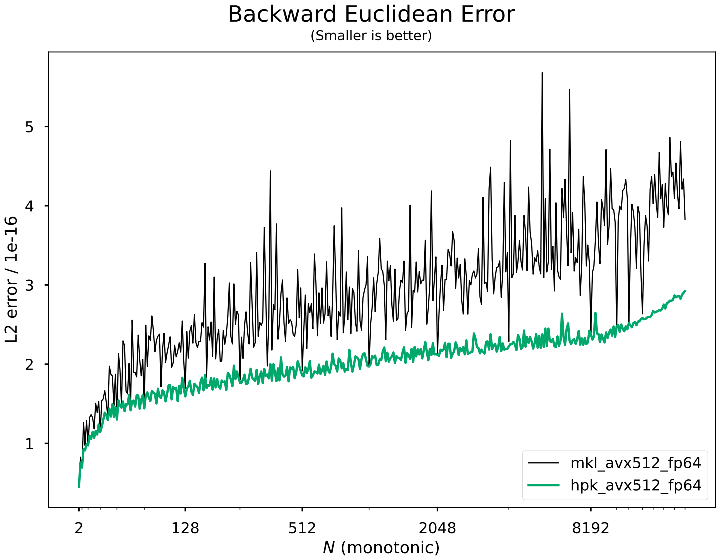
Performance: AVX512 Double Precision vs. MKL



Accuracy: AVX2 Double Precision vs. FFTW



Accuracy: AVX512 Double Precision vs. MKL



Flops and Flaps

Definition

A *floating-point operation*, or *flop*, is the addition, subtraction, or multiplication of two floating-point numbers.

Definition

A *floating-point amalgamated operation*, or *flap*, is either an FMA or a flop that is not itself a component of an FMA.

Example #1 $\quad (x_0 \cdot x_1)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0 \in \mathbb{C}$

function `cmul`

$\Re(tmp) \leftarrow \Im(x_0)\,\Im(x_1)$ `// MUL`
$\Im(tmp) \leftarrow \Im(x_0)\,\Re(x_1)$ `// MUL`

$\Re(Y_0) \leftarrow \Re(x_0)\,\Re(x_1) - \Re(tmp)$ `// FMA`
$\Im(Y_0) \leftarrow \Re(x_0)\,\Im(x_1) + \Im(tmp)$ `// FMA`

end

flops: 6
flaps: 4

Example #1 $\quad (x_0 \cdot x_1)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0 \in \mathbb{C}$

function `cmul`

$$\begin{aligned}
\Re(tmp) &\leftarrow \Im(x_0)\,\Im(x_1) && \texttt{// MUL}\\
\Im(tmp) &\leftarrow \Im(x_0)\,\Re(x_1) && \texttt{// MUL}\\
\Re(Y_0) &\leftarrow \Re(x_0)\,\Re(x_1) - \Re(tmp) && \texttt{// FMA}\\
\Im(Y_0) &\leftarrow \Re(x_0)\,\Im(x_1) + \Im(tmp) && \texttt{// FMA}
\end{aligned}$$

end

flops: 6
flaps: 4

Example #2 $\quad \left(x_0 - i\,x_1\right)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y \in \mathbb{C}$

function

$$\Re(Y) \leftarrow \Re(x_0) + \Im(x_1)$$
$$\Im(Y) \leftarrow \Im(x_0) - \Re(x_1)$$

end

flops: 2
flaps: 2

Example #2 $\quad (x_0 - i x_1)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y \in \mathbb{C}$

function

$\Re(Y) \leftarrow \Re(x_0) + \Im(x_1)$

$\Im(Y) \leftarrow \Im(x_0) - \Re(x_1)$

end

flops: 2
flaps: 2

Arm SVE `fcadd #270`

Example #2 $\left(x_0 - i\,x_1\right)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y \in \mathbb{C}$

function

$$\Re(Y) \leftarrow \Re(x_0) + \Im(x_1)$$
$$\Im(Y) \leftarrow \Im(x_0) - \Re(x_1)$$

end

Arm SVE	`fcadd #270`
x86 AVX	~~`vsubadd`~~

flops: 2
flaps: 2 ✓

Example #2 $\left(x_0 - i\,x_1\right)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y \in \mathbb{C}$

function

$\Re(Y) \leftarrow \Re(x_0) + \Im(x_1)$
$\Im(Y) \leftarrow \Im(x_0) - \Re(x_1)$

end

flops:	2
flaps:	2

✓

Arm SVE	`fcadd #270`
x86 AVX	~~`vsubadd`~~

$$e^{-2\pi i/8}x = \left(\sqrt{1/2} - \sqrt{1/2}\,i\right)x$$

Example #2 $\quad (x_0 - i\,x_1)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y \in \mathbb{C}$

function

$\Re(Y) \leftarrow \Re(x_0) + \Im(x_1)$
$\Im(Y) \leftarrow \Im(x_0) - \Re(x_1)$

end

flops: 2
flaps: 2

Arm SVE	`fcadd #270`
x86 AVX	~~`vsubadd`~~

$$e^{-2\pi i/8}x = \left(\sqrt{1/2} - \sqrt{1/2}\,i\right)x$$
$$= \sqrt{1/2}\left(1 - i\right)x$$

Example #2 $\left(x_0 - i\, x_1\right)$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y \in \mathbb{C}$

function

$$\Re(Y) \leftarrow \Re(x_0) + \Im(x_1)$$
$$\Im(Y) \leftarrow \Im(x_0) - \Re(x_1)$$

end

Arm SVE	`fcadd #270`
x86 AVX	~~`vsubadd`~~

$$\begin{aligned} e^{-2\pi i/8}\, x &= \left(\sqrt{1/2} - \sqrt{1/2}\, i\right) x \\ &= \sqrt{1/2}\left(1 - i\right) x \\ &= \sqrt{1/2}\left(x - i x\right) \end{aligned}$$

flops: 2
flaps: 2

Example #3 $\langle x_0 + x_1,\ x_0 - x_1 \rangle$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0, Y_1 \in \mathbb{C}$

function `fft_2`

$Y_0 \leftarrow x_0 + x_1$

$Y_1 \leftarrow x_0 - x_1$

end

flops: 4
flaps: 4

Example #3 $\langle x_0 + x_1,\ x_0 - x_1 \rangle$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0, Y_1 \in \mathbb{C}$

function `fft_2`
   $Y_0 \leftarrow x_0 + x_1$
   $Y_1 \leftarrow x_0 - x_1$
end

flops: 4
flaps: 4

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0, Y_1 \in \mathbb{C}$

function `fft_2`
   $Y_0 \leftarrow x_0 + x_1$ `// ADD`
   $Y_1 \leftarrow 2x_0 - Y_0$ `// FMA`
end

flops: 6
flaps: 4

Example #4 $\left\langle x_0 + \sqrt{1/2}\, x_1,\ x_0 - \sqrt{1/2}\, x_1 \right\rangle$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0, Y_1 \in \mathbb{C}$

function

$tmp \leftarrow \sqrt{1/2}\, x_1$
$Y_0 \leftarrow x_0 + tmp$
$Y_1 \leftarrow x_0 - tmp$

end

flops: 6
flaps: 6

Example #4 $\left\langle x_0 + \sqrt{1/2}\,x_1,\ x_0 - \sqrt{1/2}\,x_1 \right\rangle$

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0, Y_1 \in \mathbb{C}$

function
  $tmp \leftarrow \sqrt{1/2}\,x_1$
  $Y_0 \leftarrow x_0 + tmp$
  $Y_1 \leftarrow x_0 - tmp$
end

flops: 6
flaps: 6

input : $x_0, x_1 \in \mathbb{C}$
output: $Y_0, Y_1 \in \mathbb{C}$

function
  `// FMAs`
  $Y_0 \leftarrow x_0 + \sqrt{1/2}\,x_1$
  $Y_1 \leftarrow x_0 - \sqrt{1/2}\,x_1$
end

flops: 8
flaps: 4

$\langle X_k \rangle = \mathcal{F}\langle x_n \rangle$ for $x_0, x_1, \ldots, x_{N-1} \in \mathbb{C}$

Letting

$$\omega_N^{kn} = \exp\left(\frac{-2\pi i\, k\, n}{N}\right)$$

the Fourier transform is

$$X_k = \sum_{n=0}^{N-1} \omega_N^{kn}\, x_n$$

Radix-2 FFT

$$X_k = \sum_{n=0}^{N-1} \omega_N^{kn}\, x_n$$

$$= \sum_{n=0}^{N/2-1} \omega_N^{k(2n)}\, x_{2n} + \sum_{n=0}^{N/2-1} \omega_N^{k(2n+1)}\, x_{2n+1}$$

$$= \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn}\, x_{2n} + \omega_N^{k} \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn}\, x_{2n+1}$$

Radix-2 FFT

$$X_k = \sum_{n=0}^{N-1} \omega_N^{kn} \, x_n$$

$$= \sum_{n=0}^{N/2-1} \omega_N^{k(2n)} \, x_{2n} + \sum_{n=0}^{N/2-1} \omega_N^{k(2n+1)} \, x_{2n+1}$$

$$= \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn} \, x_{2n} + \omega_N^k \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn} \, x_{2n+1}$$

Radix-2 FFT

$$
\begin{aligned}
X_k &= \sum_{n=0}^{N-1} \omega_N^{kn}\, x_n \\
&= \sum_{n=0}^{N/2\,-1} \omega_N^{k(2n)}\, x_{2n} + \sum_{n=0}^{N/2\,-1} \omega_N^{k(2n+1)}\, x_{2n+1} \\
&= \sum_{n=0}^{N/2\,-1} \omega_{N/2}^{kn}\, x_{2n} + \omega_N^{k} \sum_{n=0}^{N/2\,-1} \omega_{N/2}^{kn}\, x_{2n+1}
\end{aligned}
$$

Radix-2 FFT

$$X_k = \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn}\, x_{2n}$$

$$+ \; \omega_N^k \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn}\, x_{2n+1}$$

Radix-4 FFT

$$X_k = \sum_{n=0}^{N/4-1} \omega_{N/4}^{kn}\, x_{4n} + \omega_N^{2k} \sum_{n=0}^{N/4-1} \omega_{N/4}^{kn}\, x_{4n+2}$$

$$+ \omega_N^{k} \sum_{n=0}^{N/4-1} \omega_{N/4}^{kn}\, x_{4n+1} + \omega_N^{3k} \sum_{n=0}^{N/4-1} \omega_{N/4}^{kn}\, x_{4n+3}$$

2,4-Split-Radix FFT

$$X_k = \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn}\, x_{2n}$$

$$+ \; \omega_N^{k} \sum_{n=0}^{N/4-1} \omega_{N/4}^{kn}\, x_{4n+1} + \omega_N^{3k} \sum_{n=0}^{N/4-1} \omega_{N/4}^{kn}\, x_{4n+3}$$

2,8-Split-Radix FFT

$$
\begin{aligned}
X_k = & \sum_{n=0}^{N/2-1} \omega_{N/2}^{kn}\, x_{2n} \\
& + \omega_N^{k} \sum_{n=0}^{N/8-1} \omega_{N/8}^{kn}\, x_{8n+1} + \omega_N^{3k} \sum_{n=0}^{N/8-1} \omega_{N/8}^{kn}\, x_{8n+3} \\
& + \omega_N^{5k} \sum_{n=0}^{N/8-1} \omega_{N/8}^{kn}\, x_{8n+5} + \omega_N^{7k} \sum_{n=0}^{N/8-1} \omega_{N/8}^{kn}\, x_{8n+7}
\end{aligned}
$$

2,8-Split-Radix FFT

function `fft_2_8<N>`

$\mathbf{u} \leftarrow$ `fft_2_8<N/2>`$(x_0, x_2, x_4, x_6, x_8, \ldots, x_{N-2})$

$\mathbf{y} \leftarrow$ `fft_2_8<N/8>`$(x_1, x_9, \ldots, x_{N-7})$

$\ldots$

for $k \leftarrow 0$ **to** $N/8 - 1$ **do**

$\ldots$

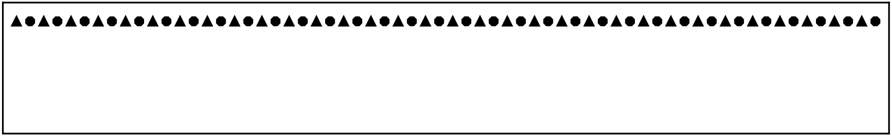
$$
\begin{aligned}
X_k &\leftarrow u_k + t_1 \\
X_{k+4N/8} &\leftarrow u_k - t_1 \\
X_{k+\ N/8} &\leftarrow u_{k+N/8} + \sqrt{1/2}\, t_9 \\
X_{k+5N/8} &\leftarrow u_{k+N/8} - \sqrt{1/2}\, t_9
\end{aligned}
$$

$\ldots$

end

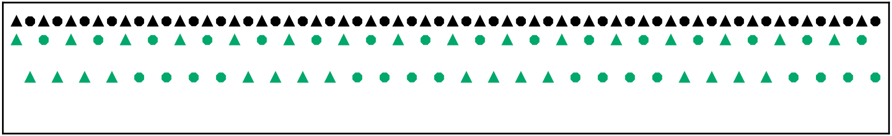
end

2,8-Split-Radix FFT



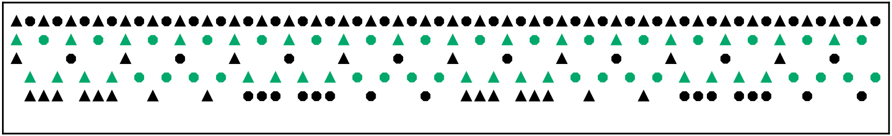
$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \ldots \right\rangle$$

2,8-Split-Radix FFT



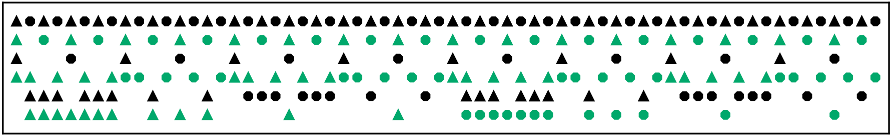
$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \dots \right\rangle$$

2,8-Split-Radix FFT



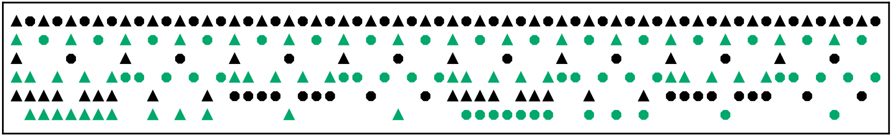
$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \ldots \right\rangle$$

2,8-Split-Radix FFT



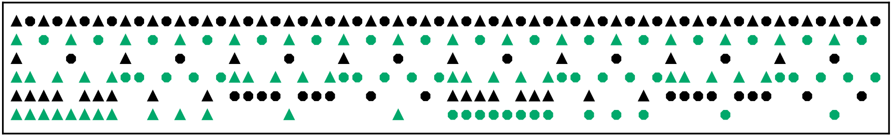
$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \ldots \right\rangle$$

2,8-Split-Radix FFT



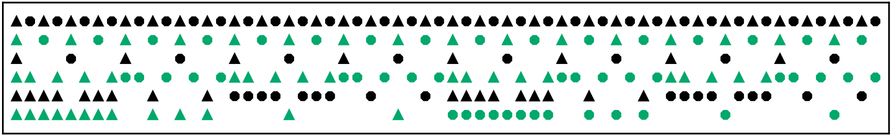
$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \ldots \right\rangle$$

2,8-Split-Radix FFT



$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \dots \right\rangle$$

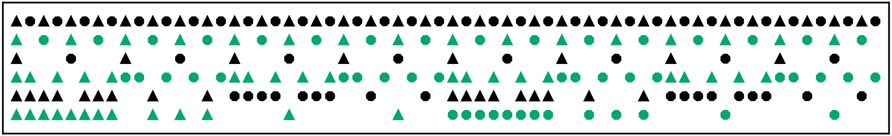
2,8-Split-Radix FFT



$$\langle g_r \rangle = \left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{5}{8}, \frac{9}{16}, \frac{13}{32}, \dots \right\rangle$$

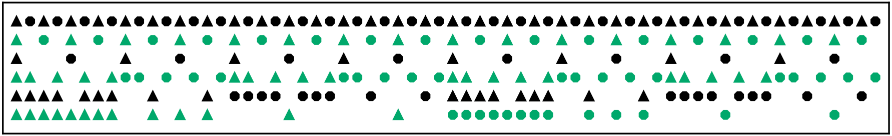
$$g_r = \frac{g_{r-1}}{2} + \frac{g_{r-3}}{2}$$

2,8-Split-Radix FFT



$$g_r = \frac{1}{2} + \left(\frac{1}{4} - \frac{\sqrt{7}}{28} i\right)\left(\frac{-1}{4} + \frac{\sqrt{7}}{4} i\right)^r + \left(\frac{1}{4} + \frac{\sqrt{7}}{28} i\right)\left(\frac{-1}{4} - \frac{\sqrt{7}}{4} i\right)^r$$

2,8-Split-Radix FFT



$$g_r = \frac{1}{2} + \left(\frac{1}{4} - \frac{\sqrt{7}}{28}i\right)\left(\frac{-1}{4} + \frac{\sqrt{7}}{4}i\right)^r + \left(\frac{1}{4} + \frac{\sqrt{7}}{28}i\right)\left(\frac{-1}{4} - \frac{\sqrt{7}}{4}i\right)^r$$

$$\lim_{r\to\infty} g_r = \frac{1}{2}$$

Asymptotic Complexity: $\mathcal{O}(N\log_2 N)$

	radix-2	radix-4	radix-8	2,4-split	2,8-split
ADD	2.0	2.0	$1.83\overline{3}$	2.0	1.75
MUL	1.0	0.75	$0.58\overline{3}$	$0.6\overline{6}$	0.5
FMA	1.0	0.75	$0.91\overline{6}$	$0.6\overline{6}$	1.0
flops	5.0	4.25	4.25	4.0	4.25
flaps	4.0	3.5	$3.3\overline{3}$	$3.3\overline{3}$	3.25

*The radix-8 alternate implementation has $4.08\overline{3}$ flops but 3.5 flaps.

**The 2,8-split-radix alternate has 4.0 flops but 3.5 flaps.

Base Case: 4-point FFT

input : $x_0, x_1, x_2, x_3 \in \mathbb{C}$
output: $X_0, X_1, X_2, X_3 \in \mathbb{C}$

function `fft_4`

$$X_0 \leftarrow (x_0 + x_2) + (x_1 + x_3)$$
$$X_1 \leftarrow (x_0 - x_2) - i(x_1 - x_3)$$
$$X_2 \leftarrow (x_0 + x_2) - (x_1 + x_3)$$
$$X_3 \leftarrow (x_0 - x_2) + i(x_1 - x_3)$$

end

Base Case: Scale 4-point FFT

input : $\alpha \in \mathbb{R}$, $x_0, x_1, x_2, x_3 \in \mathbb{C}$
output: $X_0, X_1, X_2, X_3 \in \mathbb{C}$

function `scale_fft_4`

$\hat{x}_2 \leftarrow \alpha x_2$
$X_0 \leftarrow (\alpha x_0 + \hat{x}_2) + \alpha(x_1 + x_3)$
$X_1 \leftarrow (\alpha x_0 - \hat{x}_2) - \alpha i(x_1 - x_3)$
$X_2 \leftarrow (\alpha x_0 + \hat{x}_2) - \alpha(x_1 + x_3)$
$X_3 \leftarrow (\alpha x_0 - \hat{x}_2) + \alpha i(x_1 - x_3)$

end

HPK Programming Interface

```
auto factory =
    hpk::fft::makeFactory<double>();

std::vector<std::complex<double>> v = ...

long N = std::ssize(v);

auto fft = factory->makeInplace({N});

double alpha = 0x1.0p-8;
fft->scaleForward(&alpha, v.data());
```