

```
adc_codes = [129 130 131 133 134 136 137 139 140 142 143 145 146 147 149 151  
151 153 155 156 158 159 161 162 163 165 166 167 169 170 171 173 175 175 177  
179 180 181 183 183 185 186 187 189 190 191 193 194 195 196 197 199 200 201  
202 203 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221  
222 223 223 225 226 226 227 228 229 230 230 231 232 233 233 234 235 235 236  
237 237 238 238 239 239 240 241 241 242 242 242 243 243 244 244 244 245 245  
245 246 246 246 246 247 247 247 247 247 247 247 248 248 248 248 248 248 248  
248 248 248 247 247 247 247 247 247 246 246 246 246 245 245 245 245 244 244  
244 243 243 242 242 241 241 240 240 239 239 238 237 237 236 236 235 234 234  
233 232 232 231 230 229 228 228 227 226 225 224 224 222 221 220 220 219 218  
217 216 215 214 213 212 210 210 208 208 206 205 204 203 202 200 200 198 197  
196 194 193 192 191 189 188 187 186 184 183 182 180 179 178 176 175 174 172  
171 170 168 167 165 164 162 161 160 158 157 155 154 152 151 150 148 147 145  
144 142 141 139 138 136 135 133 132 130 129 128 128 124 123 122 120 119 117  
116 114 113 112 110 108 107 106 104 102 101 100 98 97 96 94 92 91 90 88 87 86  
84 83 81 80 79 77 76 75 73 72 71 69 68 67 66 64 64 62 60 59 58 57 56 54 53 52  
51 50 49 48 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 32 31 30 29 28 27 26  
26 25 24 23 22 22 21 20 20 19 18 18 17 16 16 16 15 14 14 13 13 12 12 12 11 11  
10 10 10 9 9 9 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7  
9 9 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 19 19 20 21 22 22  
23 23 24 25 26 27 28 29 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46  
47 48 49 51 52 53 54 55 56 58 59 60 61 63 63 65 66 67 69 70 71 73 74 75 77 78  
79 81 82 83 85 86 87 89 91 92 93 95 95 98 99 101 102 103 105 106 108 109 111  
112 114 115 117 118 119 121 123 124 125 127];  
  
Tconv = 8.5e-6;  
num_values_expected = 512;  
VDD = 1.8;  
Pwr = 0.0004532;  
  
N = length(adc_codes);  
  
% FFT  
X = fft(adc_codes);  
X = X(1:N/2); % take positive frequencies only  
P = abs(X).^2; % power spectrum  
  
% Find fundamental (ignore DC bin at index 1)  
[~, fund_bin] = max(P(2:end));  
fund_bin = fund_bin + 1;  
  
% Signal power (fundamental)  
P_signal = P(fund_bin);  
  
% Noise+distortion power (everything else except DC and fundamental)  
P_noise = sum(P) - P_signal - P(1);  
  
% SNDR  
SNDR = P_signal / P_noise;  
SNDR_dB = 10*log10(SNDR)
```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Sampling Rate
fs = 1/Tconv

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    48.1480

ENOB =

    7.7057

fs =

    1.1765e+05

FOM_fJ =

    1.8453e+04

FOM_Schreier =

    129.2806

adc_codes has 512 values []

Published with MATLAB® R2025b

```

[illegible]

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Sampling Rate
fs = 1/Tconv

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    47.8336

ENOB =

    7.6534

fs =

    1.1765e+05

FOM_fJ =

    1.7711e+04

FOM_Schreier =

    129.3018

adc_codes has 512 values

```

Published with MATLAB® R2025b

```
adc_codes = [129 130 131 133 134 136 137 139 140 142 143 145 146 147 149 151  
151 153 155 156 158 159 161 162 163 165 166 167 169 170 171 173 175 175 177  
179 180 181 183 183 185 186 187 189 190 191 191 194 195 197 198 199 200 201  
202 203 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221  
222 223 223 225 226 227 227 228 229 230 231 231 232 233 234 234 235 235 236  
237 237 238 239 239 240 240 241 241 242 242 243 243 244 244 245 245 245 246  
246 246 246 247 247 247 247 247 248 248 248 248 248 248 248 248 248 248  
248 248 248 248 248 248 248 247 247 247 247 247 246 246 246 245 245 245 244  
244 244 243 242 242 242 241 241 240 240 239 238 238 237 236 236 235 234 234  
233 232 232 231 230 229 228 228 227 226 225 224 224 222 221 221 220 219 218  
217 216 215 214 213 212 210 210 208 208 206 205 204 203 202 200 200 198 197  
196 194 193 192 191 189 188 187 186 184 183 182 180 179 178 176 175 174 172  
171 170 168 167 165 164 162 161 160 158 157 155 154 152 151 150 148 146 145  
144 142 141 139 138 136 135 133 132 130 129 128 128 124 123 122 120 119 117  
116 114 113 112 110 108 107 106 104 102 101 100 98 97 96 94 92 91 90 88 87 86  
84 83 81 80 79 77 76 75 73 72 71 69 68 67 66 64 64 62 60 59 58 57 56 54 53 52  
51 50 49 48 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 32 31 30 29 28 27 26  
26 25 24 23 22 22 21 20 20 19 18 18 17 16 16 16 15 14 14 13 13 12 12 12 11 11  
10 10 10 9 9 9 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7  
9 10 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 19 19 20 21 22 22  
23 23 24 25 26 27 28 29 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46  
47 48 49 51 52 53 54 55 56 58 59 60 61 63 63 65 66 67 69 70 71 73 74 75 77 78  
79 81 82 83 85 86 87 89 91 92 93 95 95 98 99 101 102 103 105 106 108 109 111  
112 114 115 117 118 119 121 123 124 125 128];  
  
Tconv = 8.5e-6;  
num_values_expected = 512;  
VDD = 1.8;  
Pwr = 0.0004535;  
  
N = length(adc_codes);  
  
% FFT  
X = fft(adc_codes);  
X = X(1:N/2); % take positive frequencies only  
P = abs(X).^2; % power spectrum  
  
% Find fundamental (ignore DC bin at index 1)  
[~, fund_bin] = max(P(2:end));  
fund_bin = fund_bin + 1;  
  
% Signal power (fundamental)  
P_signal = P(fund_bin);  
  
% Noise+distortion power (everything else except DC and fundamental)  
P_noise = sum(P) - P_signal - P(1);  
  
% SNDR  
SNDR = P_signal / P_noise;  
SNDR_dB = 10*log10(SNDR)
```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Sampling Rate
fs = 1/Tconv

% Walden's FOM
FOM = Pwr / (2^ENOB * fs);    % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15           % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    47.3454

ENOB =

    7.5723

fs =

    1.1765e+05

FOM_fJ =

    2.0254e+04

FOM_Schreier =

    128.4751

adc_codes has 512 values []

Published with MATLAB® R2025b

```

[illegible]

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Sampling Rate
fs = 1/Tconv

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    48.4861

ENOB =

    7.7618

fs =

    1.1765e+05

FOM_fJ =

    1.7741e+04

FOM_Schreier =

    129.6207

adc_codes has 512 values []

```

Published with MATLAB® R2025b

[illegible]

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Sampling Rate
fs = 1/Tconv

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    48.4053

ENOB =

    7.7484

fs =

    1.1765e+05

FOM_fJ =

    1.9251e+04

FOM_Schreier =

    129.2255

adc_codes has 512 values

```

Published with MATLAB® R2025b

```

adc_codes = [130 131 133 134 136 137 139 140 142 143 145 146 148 149 151 152
154 155 157 158 159 161 162 164 165 167 168 169 171 172 174 175 177 178 179
181 182 183 185 186 187 189 190 191 191 194 195 196 197 199 200 201 202 203
205 206 207 208 209 210 211 213 213 215 215 217 218 219 220 221 222 223 223
225 226 226 227 228 229 230 231 231 232 233 234 235 235 236 237 237 238 239
239 240 241 241 242 242 243 243 244 244 245 245 246 246 246 247 247 247 248
248 248 249 249 249 249 250 250 250 250 250 250 250 250 250 250 250 250 250
250 250 250 250 250 250 250 250 250 249 249 249 249 248 248 248 247 247 247
246 246 245 245 244 244 244 243 242 242 241 241 240 240 239 238 238 237 236
236 235 234 233 233 232 231 230 229 228 228 227 226 225 224 223 222 221 220
219 218 217 216 215 214 213 212 211 210 208 208 206 205 204 203 202 200 199
198 197 196 194 193 192 190 189 188 186 185 184 182 181 180 178 177 176 174
173 171 170 168 168 166 164 163 162 160 160 157 156 154 153 151 150 148 147
145 144 142 141 139 138 136 135 133 132 130 129 128 126 128 123 122 120 118
117 116 114 112 112 110 108 107 105 104 102 101 100 98 96 96 94 92 91 89 88
86 85 84 82 81 80 78 77 76 74 73 72 70 69 68 66 65 64 62 61 60 59 58 56 55 54
53 52 50 49 48 47 46 45 44 43 42 40 40 38 38 36 36 35 34 33 32 31 30 29 28 28
27 26 25 24 24 23 22 21 20 20 19 19 18 17 17 16 16 15 14 14 14 13 12 12 12 11
11 10 10 10 9 9 9 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
9 9 9 10 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 18 18 19 20 20 21 22
23 23 24 25 25 26 27 28 29 30 31 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
47 47 49 50 51 52 53 55 55 57 58 59 61 62 63 65 66 67 68 70 71 72 74 75 76 78
79 80 82 83 85 86 87 89 90 91 93 95 95 97 99 100 102 103 105 106 107 109 111
111 113 115 117 118 119 121 122 124 125 128 127];

num_values_expected = 512;
VDD = 1.8;
Pwr = 0.00050205;
fs = 117.2465e3;

N = length(adc_codes);

% FFT
X = fft(adc_codes);
X = X(1:N/2); % take positive frequencies only
P = abs(X).^2; % power spectrum

% Find fundamental (ignore DC bin at index 1)
[~, fund_bin] = max(P(2:end));
fund_bin = fund_bin + 1;

% Signal power (fundamental)
P_signal = P(fund_bin);

% Noise+distortion power (everything else except DC and fundamental)
P_noise = sum(P) - P_signal - P(1);

% SNDR
SNDR = P_signal / P_noise;
SNDR_dB = 10*log10(SNDR)

```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    47.2922

ENOB =

    7.5635

FOM_fJ =

    2.2637e+04

FOM_Schreier =

    127.9654

adc_codes has 512 values []

Published with MATLAB® R2025b

```

```
adc_codes = [130 131 133 134 136 137 139 140 142 143 145 146 148 149 151 152  
154 155 157 158 159 161 162 164 165 167 168 169 171 172 173 175 177 178 179  
181 182 183 185 186 187 189 190 191 191 194 195 196 197 199 200 201 202 203  
205 206 207 208 209 210 211 213 213 215 215 217 218 219 220 221 222 223 223  
225 226 226 227 228 229 230 231 231 232 233 234 235 235 236 237 237 238 239  
239 240 241 241 242 242 243 243 244 244 245 245 246 246 246 247 247 247 248  
248 249 249 249 249 249 250 250 250 250 250 250 250 250 250 250 251 251 250  
250 250 250 250 250 250 250 250 250 249 249 249 249 248 248 248 247 247 247  
246 246 245 245 244 244 244 243 242 242 241 241 240 240 239 238 238 237 236  
236 235 234 233 233 232 231 230 229 228 228 227 226 225 224 223 222 221 220  
219 218 217 216 215 214 213 212 211 210 208 208 206 205 204 203 202 200 199  
198 197 196 194 193 191 190 189 188 186 185 184 182 181 180 178 177 176 174  
173 171 170 168 168 166 164 163 162 160 160 157 156 154 153 151 150 148 147  
145 144 142 141 139 138 136 135 133 132 130 129 128 126 128 123 128 120 118  
117 116 114 112 112 110 108 107 105 104 102 101 100 98 96 96 94 92 91 89 88  
86 85 84 82 81 80 78 77 76 74 73 72 70 69 68 66 65 64 62 61 60 59 58 56 55 54  
53 52 50 49 48 47 46 45 44 43 42 40 40 38 38 36 36 35 34 33 32 32 30 29 28 28  
27 26 25 24 24 23 22 21 20 20 19 19 18 17 17 16 16 15 14 14 14 13 12 12 12 11  
11 10 10 10 9 9 9 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8  
9 9 9 10 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 18 18 19 20 20 21 22  
23 23 24 25 25 26 27 28 29 30 31 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45  
47 47 49 50 51 52 53 55 55 57 58 59 61 62 63 65 66 67 68 70 71 72 74 75 76 78  
79 80 82 83 85 86 87 89 90 91 93 95 95 97 99 100 102 103 105 106 107 109 111  
111 113 115 117 118 119 121 122 128 125 128 127];  
num_values_expected = 512;  
VDD = 1.8;  
Pwr = 0.0004640;  
fs = 117.2025e3;  
  
N = length(adc_codes);  
  
% FFT  
X = fft(adc_codes);  
X = X(1:N/2); % take positive frequencies only  
P = abs(X).^2; % power spectrum  
  
% Find fundamental (ignore DC bin at index 1)  
[~, fund_bin] = max(P(2:end));  
fund_bin = fund_bin + 1;  
  
% Signal power (fundamental)  
P_signal = P(fund_bin);  
  
% Noise+distortion power (everything else except DC and fundamental)  
P_noise = sum(P) - P_signal - P(1);  
  
% SNDR  
SNDR = P_signal / P_noise;  
SNDR_dB = 10*log10(SNDR)
```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    44.6390

ENOB =

    7.1228

FOM_fJ =

    2.8407e+04

FOM_Schreier =

    125.6529

adc_codes has 512 values

```

Published with MATLAB® R2025b

```

adc_codes = [130 131 133 134 136 137 139 140 142 143 145 146 148 149 151 152
154 155 157 158 159 161 162 164 165 167 168 169 171 172 174 175 177 178 179
181 182 183 185 186 187 189 190 191 191 194 195 196 197 199 200 201 202 203
205 206 207 208 209 210 211 213 213 215 215 217 218 219 220 221 222 223 223
225 226 226 227 228 229 230 231 231 232 233 234 235 235 236 237 237 238 239
239 240 241 241 242 243 243 244 244 245 245 246 246 246 247 247 247 248 248
249 249 249 250 250 250 250 250 251 251 251 251 251 251 251 251 251 251 251
251 251 251 251 251 251 250 250 250 250 250 249 249 249 248 248 248 247 247
247 246 246 245 245 244 244 243 243 242 242 241 240 240 239 238 238 237 236
236 235 234 233 233 232 231 230 229 228 228 227 226 225 224 223 222 221 220
219 218 217 216 215 214 213 212 211 210 208 208 206 205 204 203 202 200 199
198 197 196 194 193 191 190 189 188 186 185 184 182 181 180 178 177 176 174
173 171 170 168 168 166 164 163 162 160 160 157 156 154 153 151 150 148 147
145 144 142 141 139 138 136 135 133 132 130 129 128 126 128 123 128 120 118
117 116 114 112 112 110 108 107 105 104 102 101 100 98 96 96 94 92 91 89 88
86 85 84 82 81 80 78 77 76 74 73 72 70 69 68 66 65 64 64 61 60 59 58 56 55 54
53 52 50 49 48 47 46 45 44 43 42 40 40 38 38 36 36 35 34 33 32 31 30 29 28 28
27 26 25 24 24 23 22 21 20 20 19 19 18 17 17 16 16 15 14 14 14 13 12 12 12 11
11 10 10 10 9 9 9 9 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
9 9 9 10 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 18 18 19 20 20 21 22
23 23 24 25 25 26 27 28 29 30 31 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
47 47 49 50 51 52 53 55 55 57 58 59 61 62 63 65 66 67 68 70 71 72 74 75 76 78
79 80 82 83 85 86 87 89 90 91 93 95 95 97 99 100 102 103 105 106 107 109 111
111 113 115 117 118 119 121 122 128 125 128 127];
num_values_expected = 512;
VDD = 1.8;
Pwr = 0.000507;
fs = 117.6255e3;

N = length(adc_codes);

% FFT
X = fft(adc_codes);
X = X(1:N/2);           % take positive frequencies only
P = abs(X).^2;          % power spectrum

% Find fundamental (ignore DC bin at index 1)
 [~, fund_bin] = max(P(2:end));
fund_bin = fund_bin + 1;

% Signal power (fundamental)
P_signal = P(fund_bin);

% Noise+distortion power (everything else except DC and fundamental)
P_noise = sum(P) - P_signal - P(1);

% SNDR
SNDR = P_signal / P_noise;
SNDR_dB = 10*log10(SNDR)

```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    44.4454

ENOB =

    7.0906

FOM_fJ =

    3.1625e+04

FOM_Schreier =

    125.0900

adc_codes has 512 values []

```

Published with MATLAB® R2025b

```

adc_codes = [130 131 133 134 136 137 139 140 142 143 145 146 148 149 151 152
154 155 157 158 159 161 162 164 165 167 168 169 171 172 174 175 177 178 179
181 182 183 185 186 187 189 190 191 191 194 195 196 197 199 200 201 202 203
205 206 207 208 209 210 211 213 213 215 215 217 218 219 220 221 222 223 223
225 226 226 227 228 229 230 231 231 232 233 234 235 235 236 237 237 238 239
239 240 241 241 242 242 243 243 244 244 245 245 246 246 246 247 247 247 248
248 248 249 249 249 249 250 250 250 250 250 250 250 250 250 250 250 250 250
250 250 250 250 250 250 250 250 249 249 249 249 249 248 248 248 247 247 246
246 246 245 245 244 244 244 243 242 242 241 241 240 240 239 238 238 237 236
236 235 234 233 233 232 231 230 229 228 228 227 226 225 224 223 222 221 220
219 218 217 216 215 214 213 212 211 210 208 208 206 205 204 203 202 200 199
198 197 196 194 193 192 190 189 188 186 185 184 182 181 180 178 177 176 174
173 171 170 168 168 166 164 163 162 160 160 157 156 154 153 151 150 148 147
145 144 142 141 139 138 136 135 133 132 130 129 128 126 128 123 122 120 118
117 116 114 112 112 110 108 107 105 104 102 101 100 98 96 96 94 92 91 89 88
86 85 84 82 81 80 78 77 76 74 73 72 70 69 68 66 65 64 62 61 60 59 58 56 55 54
53 52 50 49 48 47 46 45 44 43 42 40 40 38 38 36 36 35 34 33 32 32 30 29 28 28
27 26 25 24 24 23 22 21 20 20 19 19 18 17 17 16 16 15 14 14 14 13 12 12 12 11
11 10 10 10 9 9 9 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
9 9 9 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 18 18 19 20 20 21 22 23
23 24 25 25 26 27 28 29 30 31 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 47
47 49 50 51 52 53 55 55 57 58 59 61 62 63 63 66 67 68 70 71 72 74 75 76 78 79
80 82 83 85 86 87 89 90 91 93 95 95 97 99 100 102 103 105 106 107 109 111 111
113 115 117 118 119 121 122 124 125 128 127];

num_values_expected = 512;
VDD = 1.8;
Pwr = 0.000504;
fs = 117.190e3;

N = length(adc_codes);

% FFT
X = fft(adc_codes);
X = X(1:N/2); % take positive frequencies only
P = abs(X).^2; % power spectrum

% Find fundamental (ignore DC bin at index 1)
[~, fund_bin] = max(P(2:end));
fund_bin = fund_bin + 1;

% Signal power (fundamental)
P_signal = P(fund_bin);

% Noise+distortion power (everything else except DC and fundamental)
P_noise = sum(P) - P_signal - P(1);

% SNDR
SNDR = P_signal / P_noise;
SNDR_dB = 10*log10(SNDR)

```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    47.1004

ENOB =

    7.5316

FOM_fJ =

    2.3243e+04

FOM_Schreier =

    127.7547

adc_codes has 512 values []

Published with MATLAB® R2025b

```

```
adc_codes = [130 131 133 134 136 137 139 140 142 143 145 146 148 149 151 152  
154 155 157 158 159 161 162 164 165 167 168 169 171 172 173 175 177 178 179  
181 182 183 185 186 187 189 190 191 191 194 195 196 197 199 200 201 202 203  
205 206 207 208 209 210 211 213 213 215 215 217 218 219 220 221 222 223 223  
225 226 226 227 228 229 230 231 231 232 233 234 235 235 236 237 237 238 239  
239 240 241 241 242 242 243 243 244 244 245 245 246 246 246 247 247 247 248  
248 249 249 249 249 250 250 250 250 250 250 250 250 250 250 251 251 250  
250 250 250 250 250 250 250 250 250 249 249 249 249 248 248 248 247 247 247  
246 246 245 245 244 244 244 243 242 242 241 241 240 240 239 238 238 237 236  
236 235 234 233 233 232 231 230 229 228 228 227 226 225 224 223 222 221 220  
219 218 217 216 215 214 213 212 211 210 208 208 206 205 204 203 202 200 199  
198 197 196 194 193 191 190 189 188 186 185 184 182 181 180 178 177 176 174  
173 171 170 168 168 166 164 163 162 160 160 157 156 154 153 151 150 148 147  
145 144 142 141 139 138 136 135 133 132 130 129 128 126 128 123 128 120 118  
117 116 114 112 112 110 108 107 105 104 102 101 100 98 96 96 94 92 91 89 88  
86 85 84 82 81 80 78 77 76 74 73 72 70 69 68 66 65 64 62 61 60 59 58 56 55 54  
53 52 50 49 48 47 46 45 44 43 42 40 40 38 38 36 36 35 34 33 32 32 30 29 28 28  
27 26 25 24 24 23 22 21 20 20 19 19 18 17 17 16 16 15 14 14 14 13 12 12 12 11  
11 10 10 10 9 9 9 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7  
9 9 9 10 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 18 18 19 20 20 21 22  
23 23 24 25 25 26 27 28 29 30 31 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45  
47 47 49 50 51 52 53 55 55 57 58 59 61 62 63 65 66 67 68 70 71 72 74 75 76 78  
79 80 82 83 85 86 87 89 90 91 93 95 95 97 99 100 102 103 105 106 107 109 111  
111 113 115 117 118 119 121 122 128 125 128 127];  
  
num_values_expected = 512;  
  
VDD = 1.8;  
  
Pwr = 0.0005475;  
  
fs = 117.293e3;  
  
N = length(adc_codes);  
  
% FFT  
X = fft(adc_codes);  
X = X(1:N/2); % take positive frequencies only  
P = abs(X).^2; % power spectrum  
  
% Find fundamental (ignore DC bin at index 1)  
[~, fund_bin] = max(P(2:end));  
fund_bin = fund_bin + 1;  
  
% Signal power (fundamental)  
P_signal = P(fund_bin);  
  
% Noise+distortion power (everything else except DC and fundamental)  
P_noise = sum(P) - P_signal - P(1);  
  
% SNDR  
SNDR = P_signal / P_noise;  
SNDR_dB = 10*log10(SNDR)
```

```

% ENOB
ENOB = (SNDR_dB - 1.76) / 6.02

% Walden's FOM
FOM = Pwr / (2^ENOB * fs); % Walden FOM in Joules/conv-step
FOM_fJ = FOM * 1e15 % Convert to fJ/conv-step

% Schreier's FOM
BW = fs/2; % Nyquist bandwidth
FOM_Schreier = SNDR_dB + 10*log10(BW / Pwr)

% === Check sample count ===
if N == num_values_expected
    disp(['adc_codes has ', num2str(N), ' values []'])
else
    disp(['adc_codes has ', num2str(N), ...
        ' values [] (expected ', num2str(num_values_expected), ')'])
end

SNDR_dB =

    44.6390

ENOB =

    7.1228

FOM_fJ =

    3.3493e+04

FOM_Schreier =

    124.9376

adc_codes has 512 values []

```

Published with MATLAB® R2025b