```
% test_ddc.m
%   Tests the digital down-converter using a known test signal.
%% Block parameters
Ns = 100;   % Samples per block
n = [1:Ns];
%% System parameters
param = system_param;
%% Signal parameters
fs = param.fs; % Sample rate
f0 = param.f0; % Nominal Carrier frequency
ft = param.ft; % Tone frequency
fc = param.fc; % Symbol rate
cps = param.cps; % Cycles per symbol

% Amplitude of signal and tone
as = 1;
at = 1;

% Do repeated pattern over and over
Nframe = 100;
symb1 = [0 0 0 0 1 2 2 3 1 2 2 0 1 2 2 3 1 2 2 0 1 2 2 3 1 2 2 0 1 2 2 3];
% Replicate this frame over and Nframe times
symb = kron(ones(1, Nframe), symb1);

% Generate signal from symbols
[s s_debug] = make_signal_4psk(fs, f0, ft, cps, param.h_ps, as, at, symb);

% Compute number of blocks we have
Nb = floor(length(s)/Ns);

% Make signal into blocks
sb = reshape(s(1:Ns*Nb), Ns, Nb);

% DEBUG: Get components into blocks for debug.
% Tone signal alone
t_debug = reshape(s_debug.t(1:Ns*Nb), Ns, Nb);
% Carrier signal alone
c_debug = reshape(s_debug.c(1:Ns*Nb), Ns, Nb);
% Modulation signal alone
m_debug = reshape(s_debug.mod(1:Ns*Nb), Ns, Nb);

% DEBUG: Plot modulated signal
%for ii=1:Nb,
%  plot([1:Ns], real(m_debug(:, ii)), [1:Ns], imag(m_debug(:, ii)));
%  pause;
%end

% DEBUG: Reference design (ideal non-block operations)
```

```
%bb_ideal = conv(s.*conj(s_debug.c), param.ddc.h_lp);
%I_ideal = real(bb_ideal); I_ideal_b = reshape(I_ideal(1:Ns*Nb), Ns, Nb);
%Q_ideal = imag(bb_ideal); Q_ideal_b = reshape(Q_ideal(1:Ns*Nb), Ns, Nb);

% Process blocks
ddc_state = ddc_init(param.ddc);
str_state = str_init(param.str);

for ii=1:Nb,
  % Get a single block
  x = sb(:, ii);

  % Digital down converter
  [Ib Qb ddc_state ddc_debug] = ddc(x, ddc_state);
  [si str_state] = str(Ib,Qb, str_state);

  % DEBUG. Plot recovered carriers
  %plot(n, ddc_debug.cos1, n, real(s_debug.c((ii-1)*Ns+[1:Ns])));
  %plot(n, ddc_debug.sin1, n, imag(s_debug.c((ii-1)*Ns+[1:Ns])));

  % DEBUG. Plot I/Q Output vs. ideal conv operations
  %plot(n, Ib, n, I_ideal_b(:,ii));
  %plot(n, Qb, n, Q_ideal_b(:,ii));

  % DEBUG. Plot I/Q outputs
  sil = logical(si);
  idx = find(si);
  plot(n,Ib,'b', n, Qb, 'g', n(sil), Ib(sil), 'b*', n(sil), Qb(sil), 'g*');
  pause;
end
```