

```
function [Ib, Qb, state_out, d] = ddc(x, state_in);
% [Ib, Qb, state_out, d] = ddc(x, state_in);
% Operates on a single block of data to generate I and Q outputs.
% Inputs:
%   x   Data
% Outputs:
%   Ib, Qb  Down-converted outputs
%   d   Debug information

state = state_in;

% BPF to isolate tone
[t1 state.bpf_state] = fir(x, state.bpf_state);

% Send tone to PLL to get reference
[ref accum1 state.PLL_state] = pll(t1, state.PLL_state);

% Compensate for delay of BPF to get aligned carrier.
accum1 = accum1 + state.del;

% Generate sine and cosine waveforms. Take multiple of frequency.
accum1 = accum1 * 3;
cos1 = cos(2*pi*accum1);
sin1 = sin(2*pi*accum1);

% Do down-conversion (multiply by carrier)
i1 = cos1.*x;
q1 = -sin1.*x;

% Matched filtering
[Ib state.lpf1_state] = fir(i1, state.lpf1_state);
[Qb state.lpf2_state] = fir(q1, state.lpf2_state);

% Return debug information if desired
if (nargout >= 4),
    d.accum = accum1;
    d.cos1 = cos1;
    d.sin1 = sin1;
    d.ref = ref;
    d.t1 = t1;
end
state_out = state;
```