```
function [state_out, y] = delay(state_in, x);
% Delays a signal by the specified number of samples.
% Inputs:
%   state_in    Input state
%   x       Input buffer of samples
% Outputs:
%   state_out   Output state
%   y       Output buffer of samples

% Get input state
s = state_in;

% Copy in samples at tail
for ii=0:length(x)-1
  % Store a sample
  s.buff(s.n_t+1) = x(ii+1);
  % Increment head index (circular)
  s.n_t = bitand(s.n_t+1, s.Mmask);
end

% Get samples out from head
y = zeros(size(x));
for ii=0:length(y)-1
  % Get a sample
  y(ii+1) = s.buff(s.n_h+1);
  % Increment tail index
  s.n_h = bitand(s.n_h+1, s.Mmask);
end

% Output the updated state
state_out = s;
```