Expects transaction data to be formatted as:

0-95: sig

96-127: nonce

128-159: gasprice

160-191: value

192-223: to

224+: data

['seq', # Memory: 32...127 = sig, 128... = other data ['calldatacopy', 96, 32, ['sub', ['calldatasize'], 96]], # Load txgas to the end of data ['mstore', ['txgas'], 'msize'], # Compute sighash = sha3(nonce ++ gasprice ++ value ++ to ++ data ++ txgas ['mstore', ['sha3', 128, ['sub', 'msize', 128]], 0], # Verify signature ['call', 3000, 1, 0, 0, 128, 0, 32], ['assert', ['eq', ['mload', 0]],

], # Verify and increment nonce ['assert', ['eq', ['calldataload', 96], ['sload', 0]]], ['sstore', ['add', ['sload', 0], 1]], # Call PAYGAS ['paygas', ['calldataload', 128]], # Make the main call ['with', 'x', ['calldataload', ['sub', ['gas'], 5000], # gas ['calldataload', 192], # to ['calldataload', 160], # value 256, # data starts from (in memory) ['sub', ['msize'], 288], # data length (elide last 32 bytes as that's TXGAS) 0,
0], # Propagate return data, and success or failure ['seq', ['returndatacopy', 0, ['returndatasize']], ['if', x, ['return', 0, ['returndatasize']], ['throw', 0, ['returndatasize']]]]]]