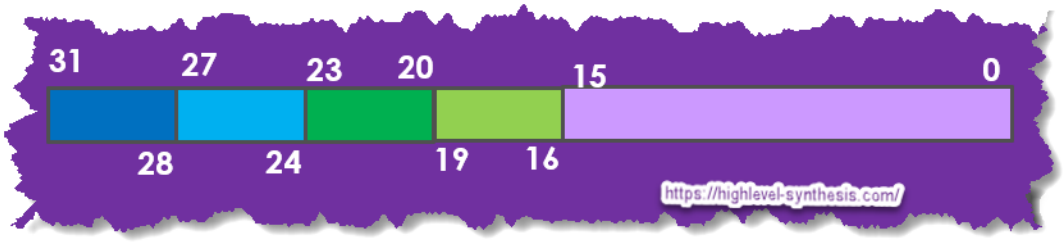
The following figure shows the format of the variable used in the algorithm.



These are the steps to write the desired code

1) First, we should define two data types

```
typedef ap_uint<32>  uint32;
typedef ap_uint<16>  uint16;
```

2) Then we should define the double_dabble function considering the data format.

```
uint32 double_dabble(uint32 scp) {
  uint16 s;
  s = scp << 1;
  if (s(19, 16) > 4)
    s(19, 16) = s(19, 16) + 3;
  if (s(23, 20) > 4)
    s(23, 20) = s(23, 20) + 3;
  if (s(27, 24) > 4)
    s(27, 24) = s(27, 24) + 3;
  return s;
}
```

3) The we should write the binary to BCD function and 15 times call the double_dabble.

```
void binary2bcd(uint16 in_binary,
                          uint16 *packed_bcd) {
#pragma HLS INTERFACE ap_none port=in_binary
#pragma HLS INTERFACE ap_none port=packed_bcd
#pragma HLS INTERFACE ap_ctrl_none port=return

  uint32 scratch_pad = in_binary;

  scratch_pad = double_dabble(scratch_pad);
  scratch_pad = double_dabble(scratch_pad);
  scratch_pad = double_dabble(scratch_pad);
  scratch_pad = double_dabble(scratch_pad);
  …
  scratch_pad = double_dabble(scratch_pad);
  scratch_pad = double_dabble(scratch_pad);
  scratch_pad = scratch_pad << 1;

  *packed_bcd   = scratch_pad(31, 16);
}
```

15 times

https://highlevel-synthesis.com/