The degree $m$ of the CRC polynomial can be chosen up to a maximum value of 32. In case $m < 32$, a calculated CRC value will be located in the $m$ most significant bits of the CRC register (CRCC), while all other bits will be 0.

\[
\text{CRCC: } \begin{array}{cccccc}
31 & 30 & 32-m & 31-m & 0 & \\
\end{array}
\]

The width $w$ of the data to be processed in parallel is limited to 32. For $w < 32$, data is expected at the most significant bit positions of the data bus. In case $w < m$, it needs to be made sure that data bits $32-m$ to $31-w$ are all set to 0 as illustrated in the following:

\[
\text{DATA: } \begin{array}{cccccc}
31 & 30 & 32-w & 31-w & 32-m & 31-m & 0 & \\
\end{array}
\]

In order to set up the CRC module to a specific polynomial and data width, 32 configuration words need to be supplied. The following matlab code generates these values:

```matlab
function FM = getFM(P,m,w)
P=[zeros(m-length(P),1); P]
if (m>32 || w > 32)
    error('Polynomial degree and data width must not exceed 32');
end
F= [[P; zeros(32-m,1)], [eye(31);zeros(1,31)]];
res = F;
for i=1:1:w-1
    res = mod(res*F,2);
end
res(1:32,max(m,w)+1:32)=0; % adjustment for w<32 && m<32
nbit=2.^((size(res,2)-i-1:0));
FM=dec2hex(nbit*res.');
```

For the CRC32 Ethernet polynomial 0x04C11DB7 and a data width of 32, the configuration values can be obtained with:

```matlab
getFM sscanf(dec2bin(hex2dec('4C11DB7')),'%1d'),32,32)
```
Output:

```
ans =
FB808B20
7DC04590
BEE022C8
5F701164
2FB808B2
97DC0459
BDE890DC
58374486
AC1BA243
AD8D5A01
AD462620
56A31310
2B518988
95A8C4C4
CAD46262
656A3131
493593B8
249AC9DC
924D64EE
C926B277
9F13D21B
B409622D
21843A36
9C0C211D
33E185AD
627049F6
313824FB
E31C995D
8A0E78BE
C58763C7
19033AC3
F7011641
```

These values need to be written into the DMA controller address block starting at offset `0x0180`. Consequently, the last value of the list corresponds to offset `0x01FC`.

- The CRC circuit can be also configured to calculate several smaller CRCs for corresponding data chunks, i.e. two independent CRC16 for the each halfword of the data. This option still needs to be incorporated into the `getFM` function.