113:AS
crc_poly_size_reversed[3] = crc_poly_size[CRC_SIZE - 2 - 3];
111:AS
crc_poly_size_reversed[1] = crc_poly_size[CRC_SIZE - 2 - 1];
112:AS
MASK_REVERSED[1] = MASK[CRC_SIZE - 2 - 4];
124:AS
MASK_REVERSED[2] = MASK[CRC_SIZE - 2 - 5];
125:AS
crc_poly_size_reversed[6] = crc_poly_size[CRC_SIZE - 2 - 6];
116:AS
MASK_REVERSED[3] = MASK[CRC_SIZE - 2 - 1];
118:AS
MASK_REVERSED[4] = MASK[CRC_SIZE - 2 - 2];
114:AS
MASK_REVERSED[5] = MASK[CRC_SIZE - 2 - 0];
113:AS
crc_poly_size_reversed[0] = crc_poly_size[CRC_SIZE - 2 - 0];
123:AS
crc_poly_size_reversed[5] = crc_poly_size[CRC_SIZE - 2 - 5];
120:AS
MASK_REVERSED[6] = MASK[CRC_SIZE - 2 - 3];
117:AS
crc_poly_size_reversed[2] = crc_poly_size[CRC_SIZE - 2 - 2];
121:AS
crc_poly_size_reversed[4] = crc_poly_size[CRC_SIZE - 2 - 4];
126:AS
MASK_REVERSED[0] = MASK[CRC_SIZE - 2 - 0];
129:AS
feedback = crc_poly & (CRC_SIZE{ crc_in[CRC_SIZE - 1] * data_in });
130:AS
crc_in_masked = crc_in[CRC_SIZE-2:0] & ~(crc_poly_size_reversed[CRC_SIZE-2:0] & MASK_REVERSED);
131:AS
crc_out = { crc_in_masked ^ feedback[CRC_SIZE-1:1], feedback[0] };