**CHECKSUM calculation verification**

The NIOS II processor is servicing the UART interrupt, checking for the beginning of a packet, which consists of 32 bytes, in the following form

|  |  |  |
| --- | --- | --- |
| byte order | byte value | description |
| 0 | 0x20 | sync\_0 |
| 1 | 0x40 | sync\_1 |
| 2 | 0xXX | CH1 low |
| 3 | 0x03 to 0x07 | CH1 high |
| 4 | 0xXX | CH2 low |
| 5 | 0x03 to 0x07 | CH2 high |
| 6 | 0xXX | CH3 low |
| 7 | 0x03 to 0x07 | CH3 high |
| 8 | 0xXX | CH4 low |
| 9 | 0x03 to 0x07 | CH4 high |
| 10 | 0xXX | CH5 low |
| 11 | 0x03 to 0x07 | CH5 high |
| 12 | 0xXX | CH6 low |
| 13 | 0x03 to 0x07 | CH6 high |
| 14 | 0xXX | CH7 low |
| 15 | 0x03 to 0x07 | CH7 high |
| 16 | 0xXX | CH8 low |
| 17 | 0x03 to 0x07 | CH8 high |
| 18 | 0xXX | CH9 low |
| 19 | 0x03 to 0x07 | CH9 high |
| 20 | 0xXX | CH10 low |
| 21 | 0x03 to 0x07 | CH10 high |
| 22 | 0xDC | constant |
| 23 | 0x05 | constant |
| 24 | 0xDC | constant |
| 25 | 0x05 | constant |
| 26 | 0xDC | constant |
| 27 | 0x05 | constant |
| 28 | 0xDC | constant |
| 29 | 0x05 | constant |
| 30 | 0xXX | CHECKSUM |
| 31 | 0xFE | sync\_end |

source code to calculate the checksum which tests whether the received packet is correct

**while**(1) {

// uart

**if**(!EmptyUart1()) {

// new char received

previous\_char = current\_char;

current\_char = GetUart1();

char\_position++;

// detect start of the packet

**if**( (previous\_char == 0x20) && (current\_char == 0x40) ){

char\_position = 1;

checksum = 0xDF;

};

// current\_char inside packet

**if**( (char\_position>=1) && (char\_position<=29) ){

// calculate XOR

checksum = checksum + (~current\_char) + 1;

// store in memory

// IOWR(SHARED\_MEM\_BASE, char\_position, current\_char)

}

// checksum

**if**( (char\_position==30) ){

**if** (checksum == current\_char){

alt\_printf("-");

}**else**{

alt\_printf("X");

}

i++;

}

**if**(i == 128){

alt\_printf("\n");

i = 0;

}

}

} //while

program output

- means checksum matches the one in the packet

X means calculated checksum is incorrect

