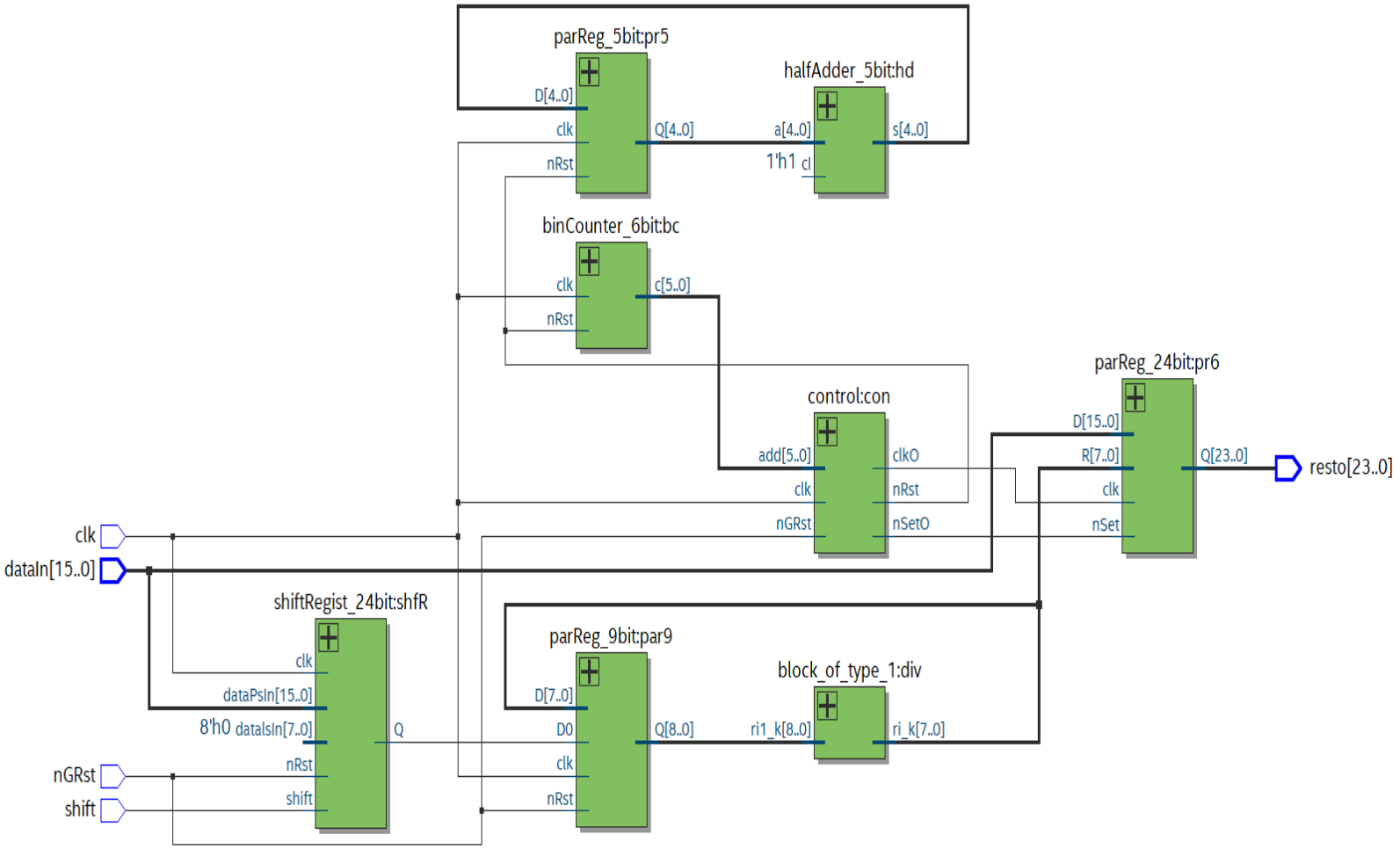
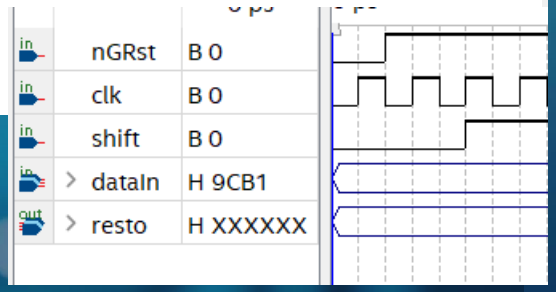
CRC8_T3G3

FEITO POR: ALDEBERTO ROSÁRIO 105589

RUI LAMEIRAS 102817

ENCODER'S BIT SIRIAL IMPLEMENTATION





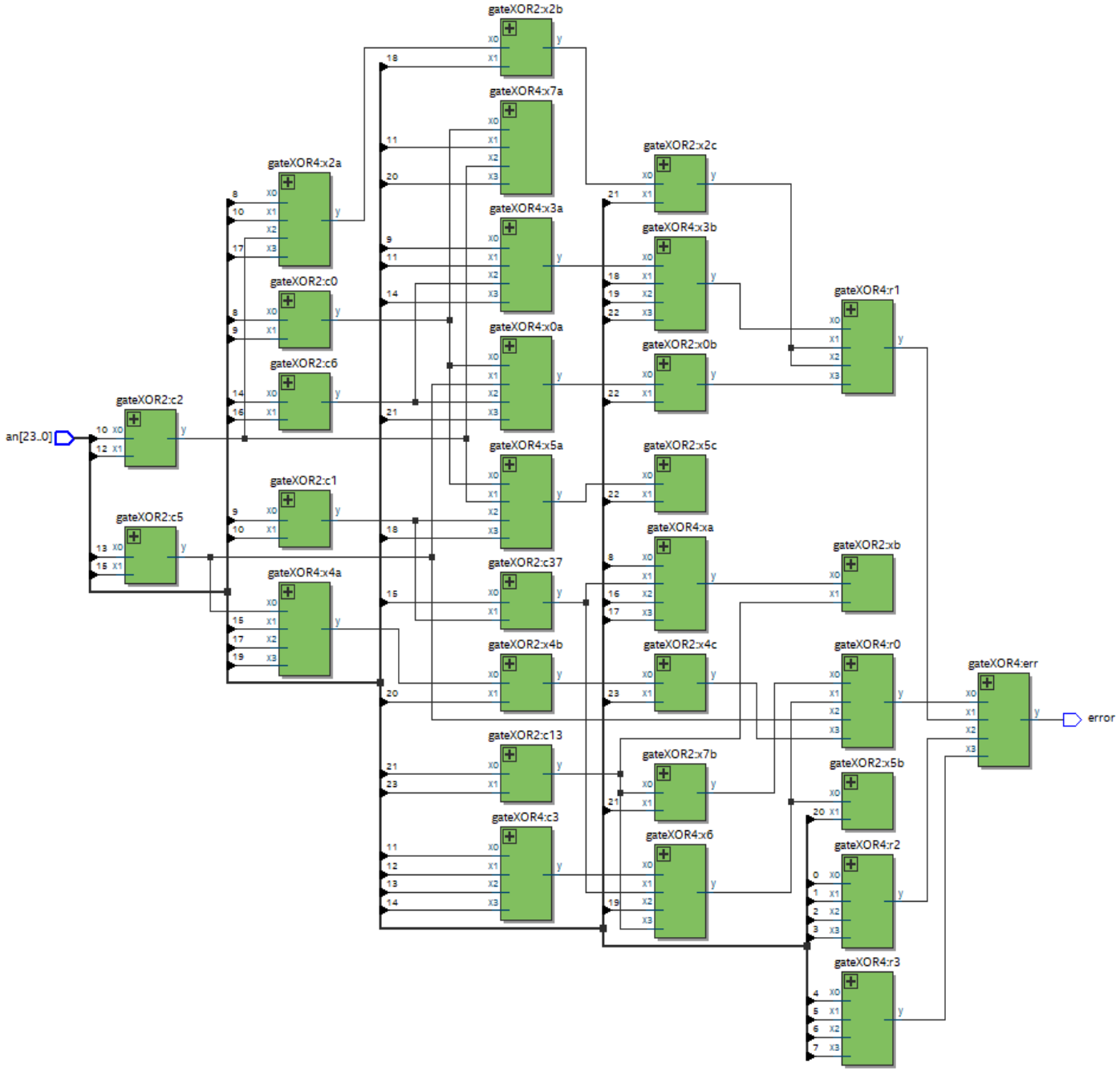
It takes 25 clock cycles to show the end value

To run the simulation (it must look like the image):

- Reset = 1after 1 cicly of clock;
- Clock: 40 ns, duty cicly = 50 with set end time: 2 us;
- The shift must be at zero in the first active clock cycle, then it must be set at 1.
- DataIn is the dividend;
- Resto is the result

ENCODER'S BIT SIRIAL IMPLEMENTATION(LOGIC)

Our idea is to use a 24-bit parallel shift register to send one bit (the most significant bit) at a time to the 9-bit parallel register. Then this bit will be sent to the block_of_type_1, where the division will be done. (We always do the division, we just don't print the value, instead it will print FFFF (Value that indicates that we are still processing data)). After division, the bit will return to the parallel register and will be shifted right by 1 while we receive the next bit. At some point in the future we will have finally finished division (clock cycle 25, to be precise). So the control, upon recognizing that it is in that cycle, will print that value. Then it will continue the division but print the data processing value (FFFFFF) instead of the division value. Blocks that are not part of the Encoder: 24-bit shift register, 9-bit parallel register and 24-bit parallel register.



CHECKER BIT PARALLEL IMPLEMENTATION

It as a total cost of 43 xor's

3 x-or propagation time delays in the worst case

CHECKER BIT PARALLEL IMPLEMENTATION(LOGIC)

We used the properties of the remainder to create this version, previously 54 xors were required and with 9 xors worst case propagation time delays. After simplification, we are left with 43 xors and 3 xors propagation time delays. We chose to do it this way because we concluded that it would be better to shorten the size of the circuit (only 6 layers) and have 3 xors propagation time, than to have a long circuit in which we could have twice the layer, where each one would have 2 xors delays propagation time.

$a0 \oplus a1 = c0$; $a1 \oplus a2 = c1$; $a2 \oplus a4 = c2$; $a3 \oplus a4 \oplus a5 \oplus a6 = c3$;

$C3 \oplus a7 = c37$; $a5 \oplus a7 = c5$; $a6 \oplus a8 = c6$; $a13 \oplus a15 = c13$