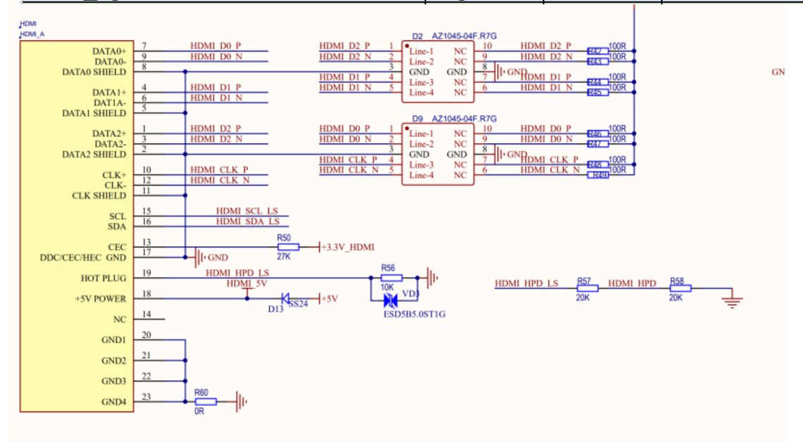
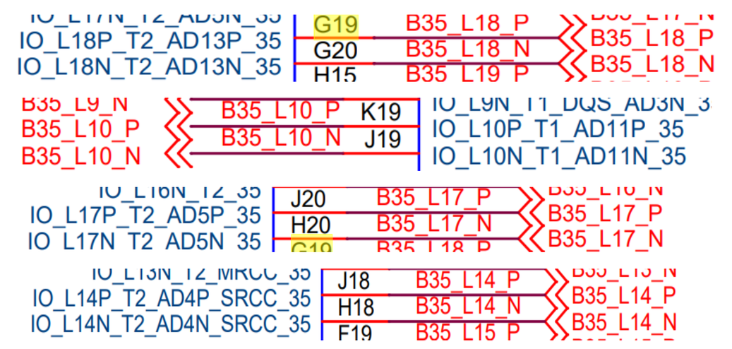
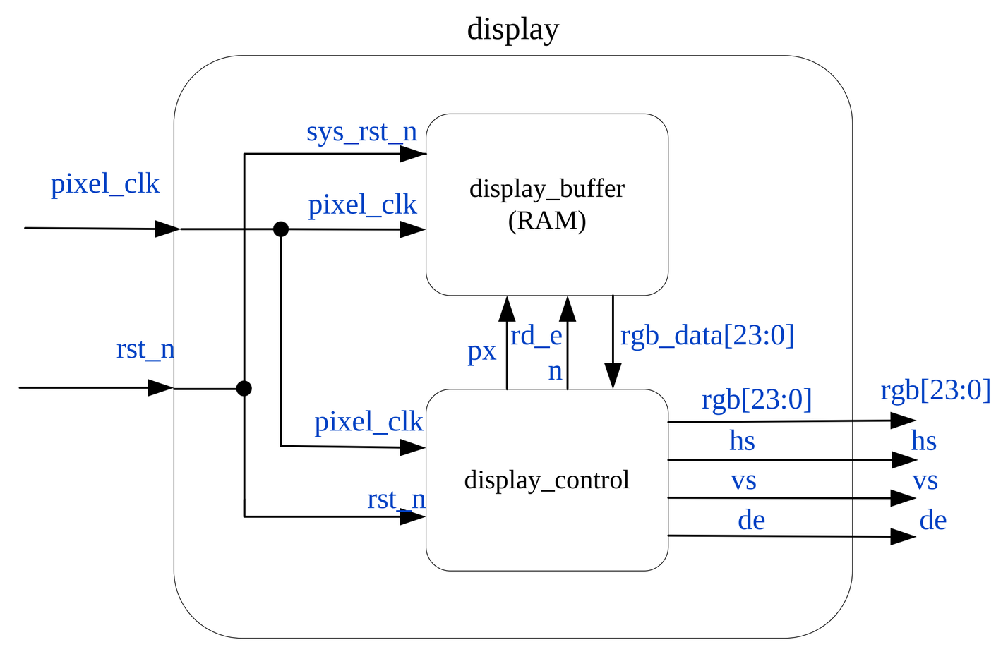
hardware

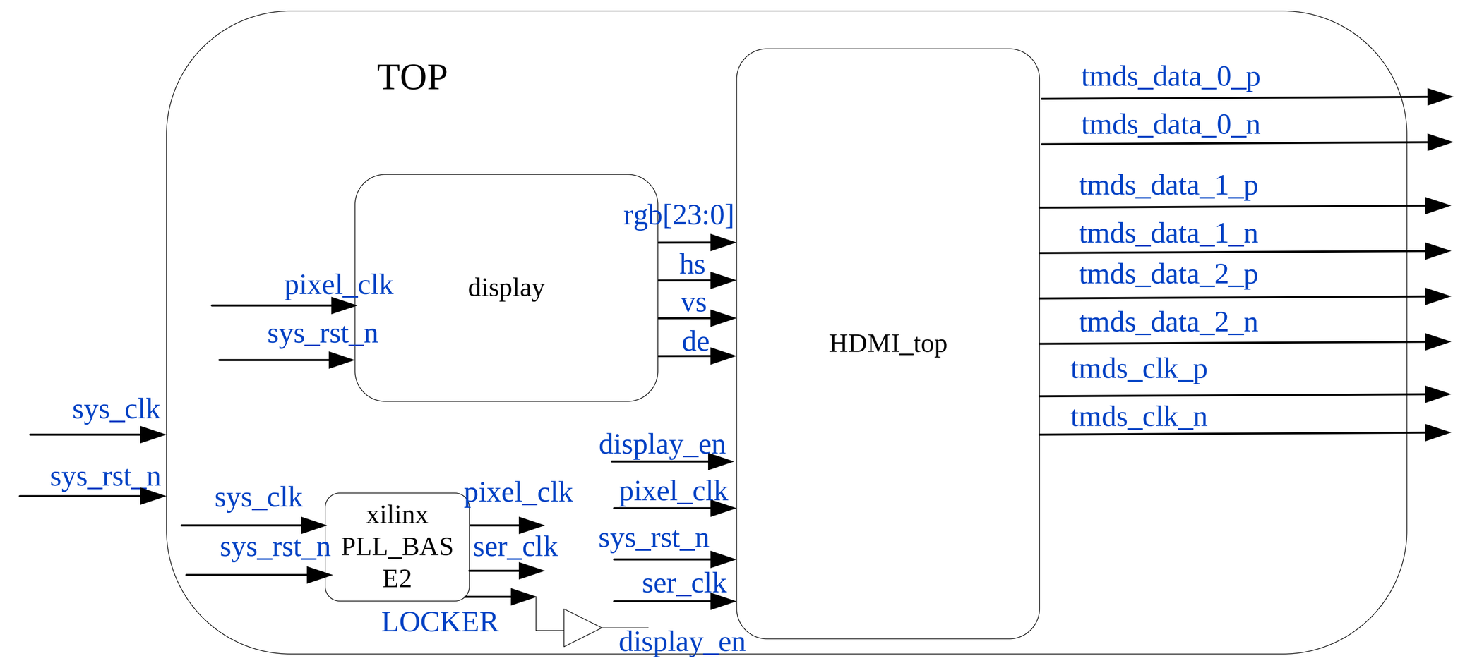
HDMI接口			
tmds_data_p[0]	output	G19	HDMI的DATA0通道的P端
tmds_data_p[1]	output	K19	HDMI的DATA1通道的P端
tmds_data_p[2]	output	J20	HDMI的DATA2通道的P端
tmds_clk_p	output	J18	HDMI的CLK通道的P端
tmds_scl	output	R19	HDMI的SCL信号
tmds_sda	output	P20	HDMI的SDA信号
tmds_hpd	input	L19	HDMI的热插拔信号

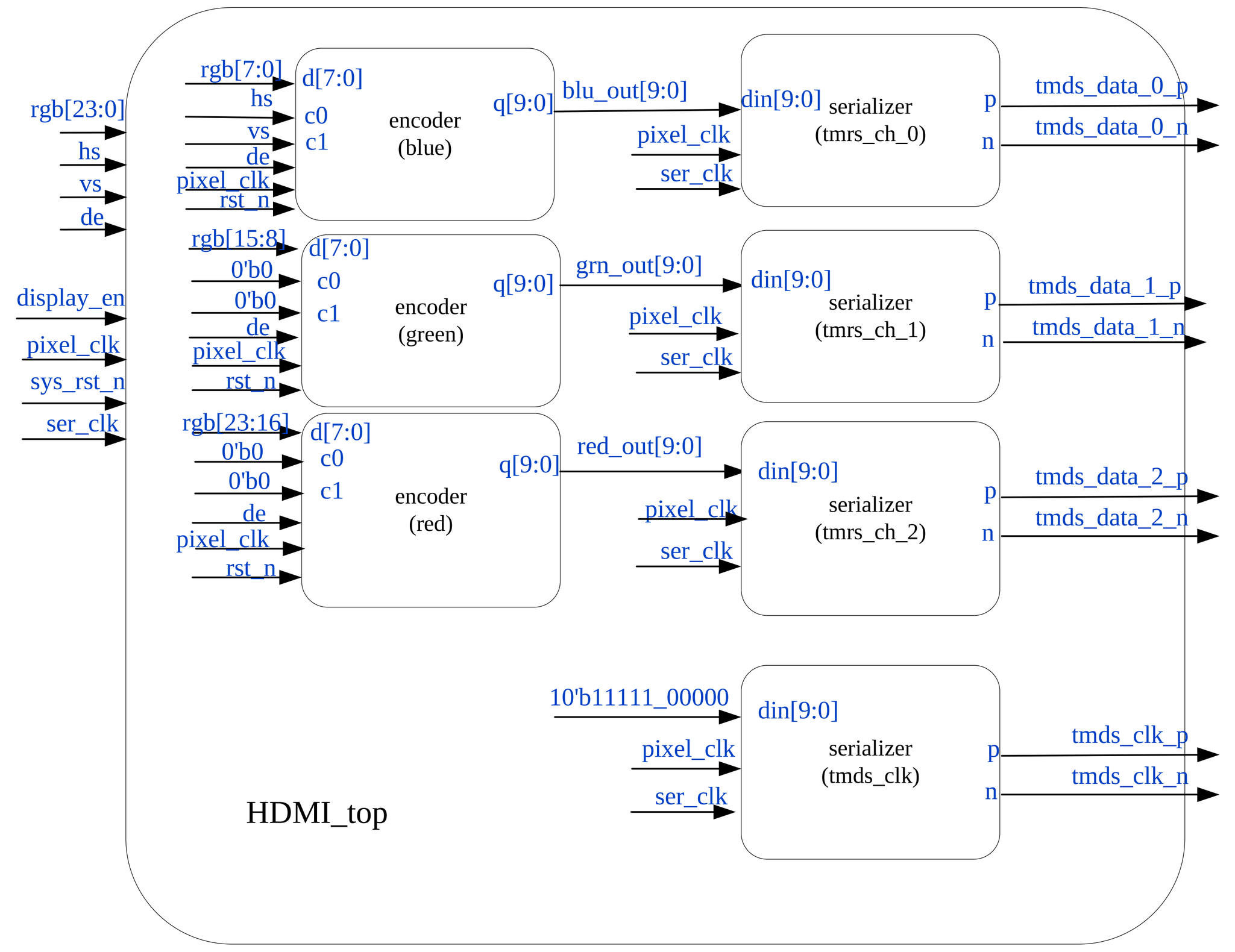


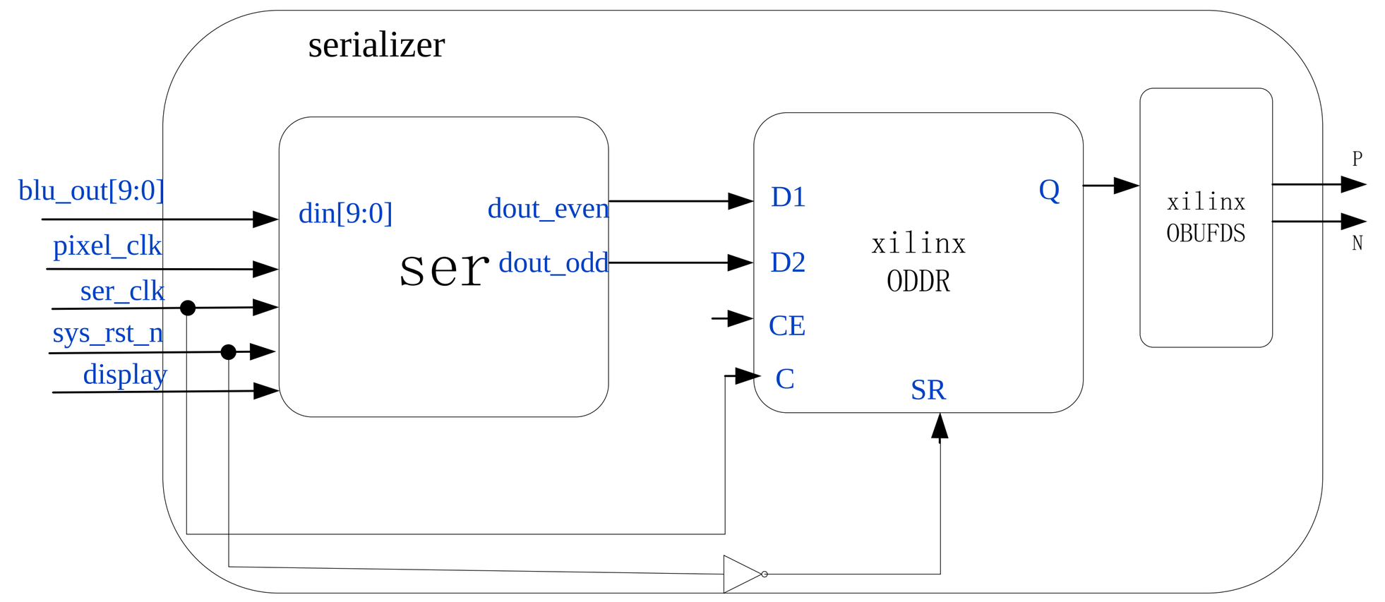




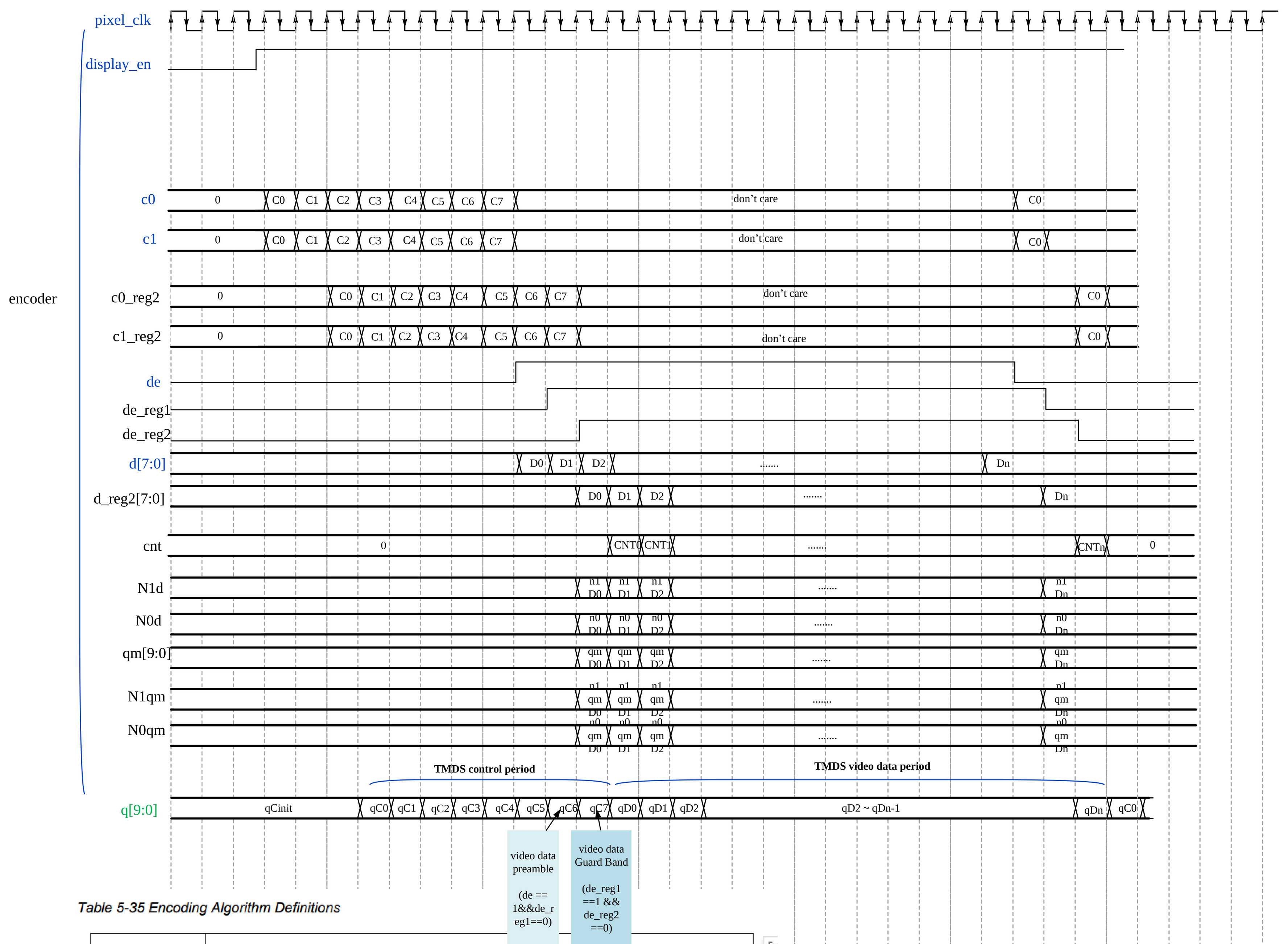
design blocks







design waveform



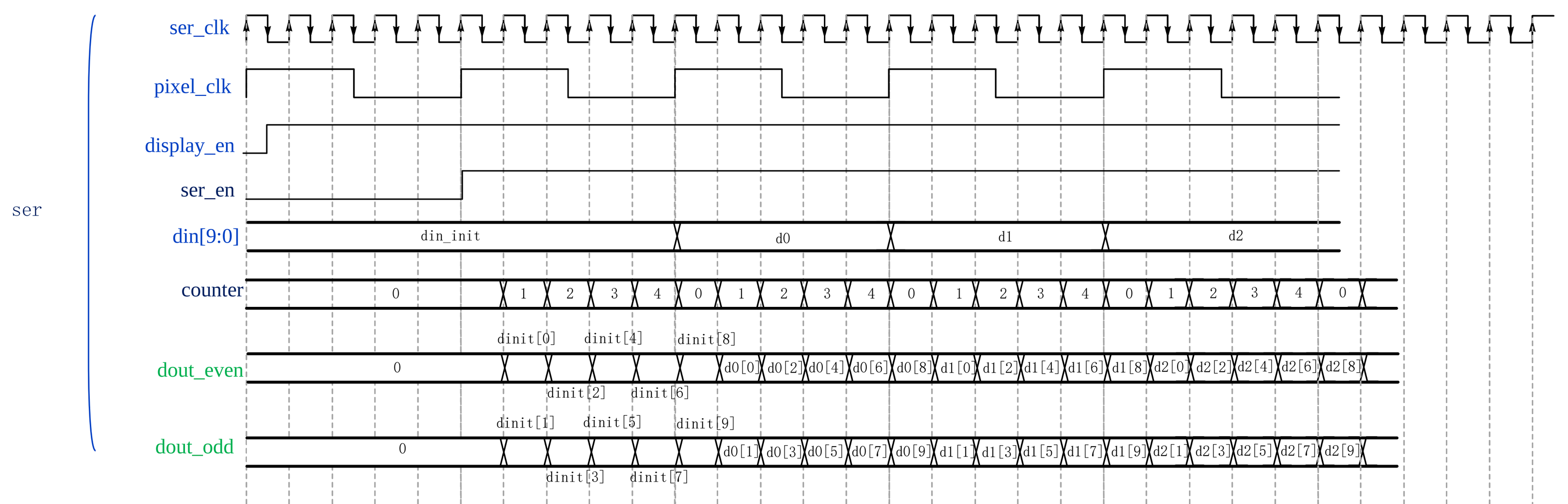


Table 5-35 Encoding Algorithm Definitions

D	The encoder input data set. D is 8-bit pixel data
cnt	This is a register used to keep track of the data stream disparity. A positive value represents the excess number of "1"s that have been transmitted. A negative value represents the excess number of "0"s that have been transmitted. The expression cnt{t-1} indicates the previous value of the disparity for the previous set of input data. The expression cnt(t) indicates the new disparity setting for the current set of input data.
q_m	Intermediate value.
q_out	These 10 bits are the encoded output value.
$N_1\{x\}$	This operator returns the number of "1"s in argument "x"
$N_0\{x\}$	This operator returns the number of "0"s in argument "x"

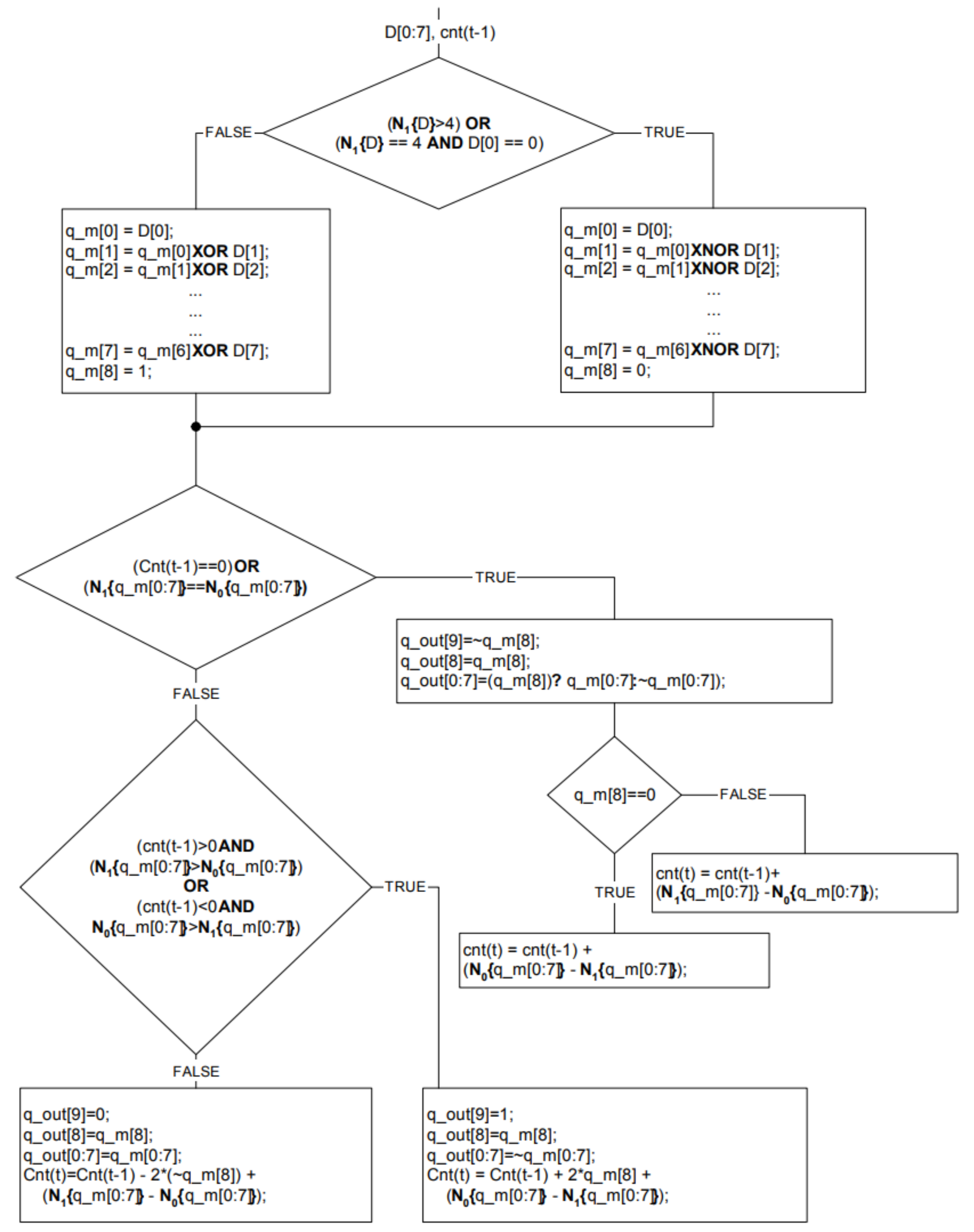


Figure 5-7 TMDS Video Data Encode Algorithm

Table 5-2 Preambles for Each Data Period Type

CTL0	CTL1	CTL2	CTL3	Data Period Type
1	0	0	0	Video Data Period
1	0	1	0	Data Island Period

5.4.2 Control Period Coding

Each TMDS channel has two control signals, which are encoded into 10 bits during Co Periods. For each of the three channels these signals are shown in Table 5-34.

Table 5-34 Control-signal Assignment

TMDS Channel	D0	D1
0	HSYNC	VSYNC
1	CTL0	CTL1
2	CTL2	CTL3

The two Control signals for each of the three TMDS channels are encoded as follows:

```
case (D1, D0):
      0, 0:  q_out[9:0] = 0b1101010100;
      0, 1:  q_out[9:0] = 0b0010101011;
      1, 0:  q_out[9:0] = 0b0101010100;
      1, 1:  q_out[9:0] = 0b1010101011;
endcase;
```

5.2.2.1 Video Guard Band

Table 5-5 Video Leading Guard Band Values

```
case (TMDS Channel Number):
      0:  q_out[9:0] = 0b1011001100;
      1:  q_out[9:0] = 0b0100110011;
      2:  q_out[9:0] = 0b1011001100;
endcase
```