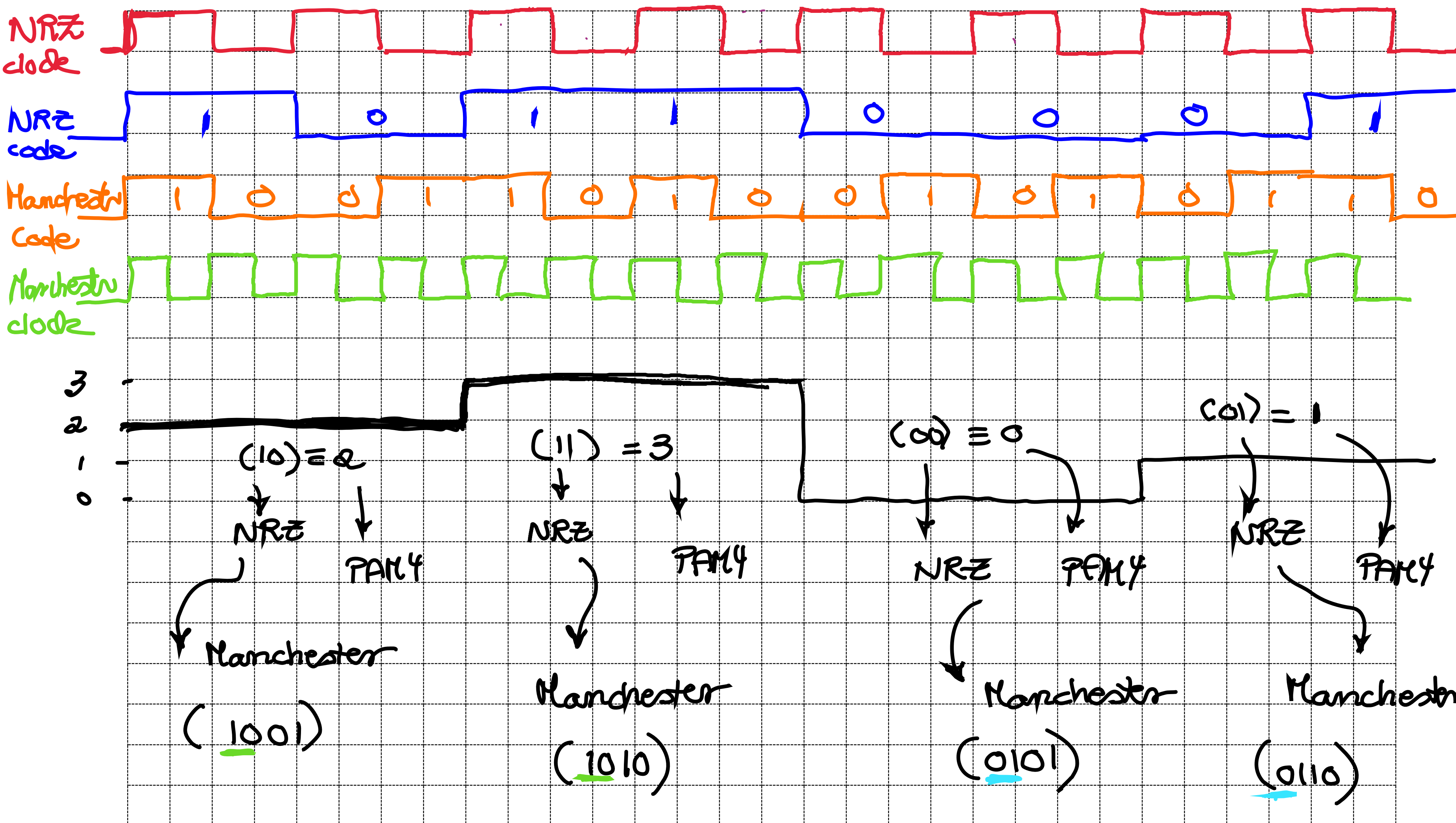


NRZ clock
NRZ code
1 0 1 1 0 0 0 1
Manchester code
1 0 0 1 1 0 1 0 0 1 0 1 0 1 1 0
Manchester clock
3
2
1
0
(10)=2
NRZ
PAM4
Manchester
(1001)
(11) =3
NRZ
PAM4
Manchester
(1010)
(00) ≡ 0
NRZ
PAM4
Manchester
(0101)
(01) = 1
NRZ
PAM4
Manchester
(0110)

A few points to think about:

PAM4 combines two NRZ bits, which means 4 Manchester bits.
Manchester 0101 = 00 NRZ = 0 PAM4
Manchester 0110 = 01 NRZ = 1 PAM4
Manchester 1001 = 10 NRZ = 2 PAM4
Manchester 1010 = 11 NRZ = 3 PAM4

This means that:

- PAM4 will change its value every 4 clock cycles from the Manchester clk.
- Some sequences are not possible if we go from Manchester to PAM4. (For example, 1111 is not possible, 0000 is not possible.. Only the ones given above are possible) So, some states will only have one outward arrow representing only one possible input. (either 0 or 1). If I start with a 0, we are 100% sure the next bit will be 1; if we start the sequence with 1, then 100% the following bit will be a 0.
- The PAM4 level output is always given using two output bits to give you the 4 possible levels.

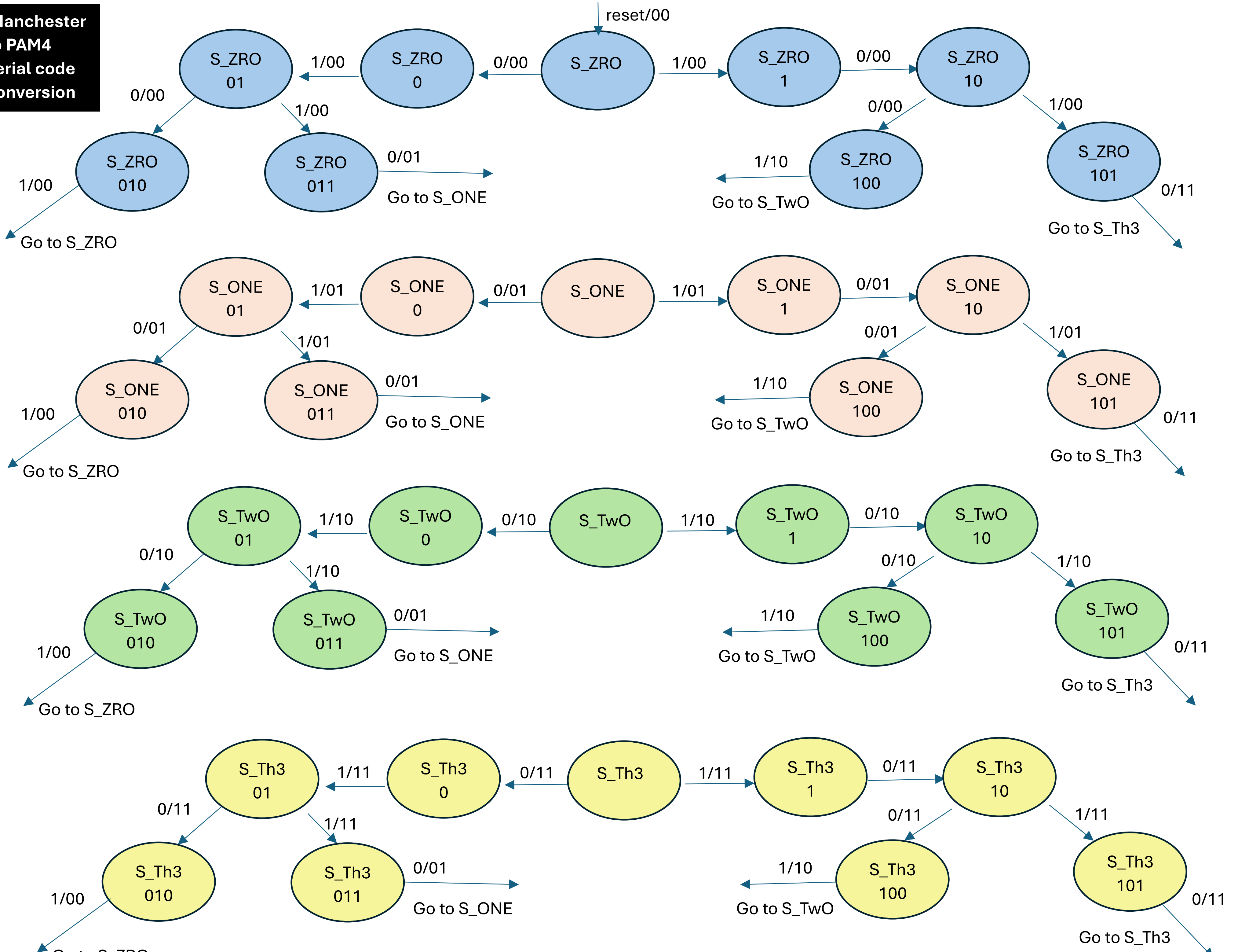
In the simulation of the Manchester to PAM4 code line converter, the PAM4 value will be delayed by 4 clock cycles (not perfectly aligned as in the timing diagram shared) because we need to wait for the serially incoming bit to figure out what the PAM4 level will be.

We did a 4Bit sequence detector as one of the examples for the FSM. We had one output that went high when a sequence was detected and was zero otherwise. The main difference here is that we need to latch the PAM4 output over 4 clock cycles, until we receive the next 4 bits of Manchester sequence. This will increase the number of states that we need significantly. (36 states)

This will require that we develop a strategy to keep our code readable, organized, parametrized to be able to troubleshoot and debug as needed.

We will learn later how we can use registers and split the design into datapath and controller to have an efficient converter design with serial input.

Manchester to PAM4 serial code conversion



For assigning numbers to the states, you can choose numbers that make it easy to track and verify in the simulation results. For example:

```
//   parameter     state_name    = 8'b_PAM4level_ManchesterCodeReceived

parameter    S_ZRO         = 8'b_0000_0101;   //  HEX  05   the last bit of the Manchester 0101 has
                                              //            been received and the output is at 00 level,
parameter    S_ZRO_xxx0  = 8'b_0000_0000;   //  HEX  00   the first bit of a new Manchester code is 0
                                              //            and the last output is latched at 00 level.
parameter    S_ZRO_xx01   = 8'b_0000_0001;   //  HEX  01   Two bits of a new Manchester code are 01
                                              //            and the output is latched at 00 level.
parameter    S_ZRO_x010  = 8'b_0000_0010;   //  HEX  02   3 bits of a new Manchester code are 010
                                              //            and the output is latched at 00 level.
parameter    S_ZRO_x011  = 8'b_0000_0011;   //  HEX  03   3 bits of a new Manchester code are 011
                                              //            and the output is latched at 00 level.
parameter    S_ZRO_xxx1   = 8'b_0000_1111;   //  HEX  0F   the first bit of a new Manchester code is 1
                                              //            and the output is latched at 00 level.
parameter    S_ZRO_xx10   = 8'b_0000_1110;   //  HEX  0E   Two bits of a new Manchester code is 10
                                              //            and the output is latched at 00 level.
parameter    S_ZRO_x101   = 8'b_0000_1101;   //  HEX  0D   3 bits of a new Manchester code is 101
                                              //            and the output is latched at 00 level.
parameter    S_ZRO_x100   = 8'b_0000_1100;   //  HEX  0C   3 bits of a new Manchester code is 100
                                              //            and the output is latched at 00 level.

parameter    S_ONE         = 8'b_0000_0110;   //  HEX  16   the last bit of the Manchester 0101 has
                                              //            been received and the output is at 01 level,
parameter    S_ONE_xxx0  = 8'b_0001_0000;   //  HEX  10   the first bit of a new Manchester code is 0
                                              //            and the last output is latched at 01 level.
parameter    S_ONE_xx01   = 8'b_0001_0001;   //  HEX  11   Two bits of a new Manchester code are 01
                                              //            and the output is latched at 01 level.
parameter    S_ONE_x010  = 8'b_0001_0010;   //  HEX  12   3 bits of a new Manchester code are 010
                                              //            and the output is latched at 01 level.
parameter    S_ONE_x011  = 8'b_0001_0011;   //  HEX  13   3 bits of a new Manchester code are 011
                                              //            and the output is latched at 01 level.
parameter    S_ONE_xxx1   = 8'b_0001_1111;   //  HEX  1F   the first bit of a new Manchester code is 1
                                              //            and the output is latched at 01 level.
parameter    S_ONE_xx10   = 8'b_0001_1110;   //  HEX  1E   Two bits of a new Manchester code is 10
                                              //            and the output is latched at 01 level.
parameter    S_ONE_x101   = 8'b_0001_1101;   //  HEX  1D   3 bits of a new Manchester code is 101
                                              //            and the output is latched at 01 level.
parameter    S_ONE_x100   = 8'b_0001_1100;   //  HEX  1C   3 bits of a new Manchester code is 100
                                              //            and the output is latched at 01 level.

parameter S_TwO = 8'b0010_1001;        parameter S_Th3 = 8'b0011_1010;       ... and so on
```