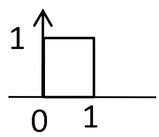
EE 5501 Prof. Jindal
Digital Communication Dec. 19, 2009

Final Exam

The exam has 4 questions, for a total of 100 points. For all problems except the first, you must show your work to receive credit.

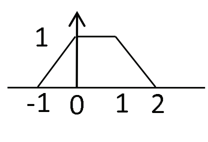
1. **Nyquist Criterion** (10 pts)
   For each of the following waveforms, indicate whether it satisfies the Nyquist criterion, and whether it is a square-root Nyquist pulse. The symbol period is $T = 2$. For each part circle your answers. You do not need to show your work for this question.

   (a)



Nyquist: Yes No Square-root Nyquist: Yes No
(5 pts)

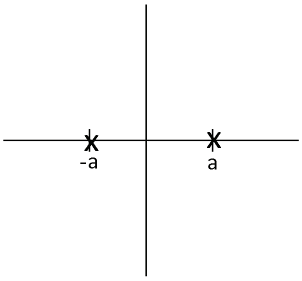
   (b)



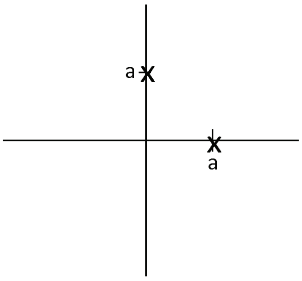
Nyquist: Yes No Square-root Nyquist: Yes No
(5 pts)

2. **Constellations** (20 pts)
In each of the following parts, you are given two constellations (the constellation points are indicated by X's). Determine which of the two constellations has a *larger* d_{min}, assuming that the constellations are scaled to have the *same energy per-bit*.

(a) (4 pts)

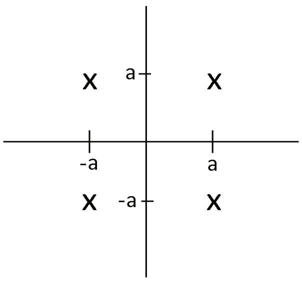


Constellation 1

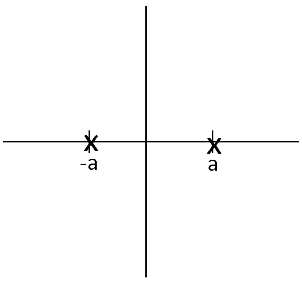


Constellation 2

(b) (4 pts)

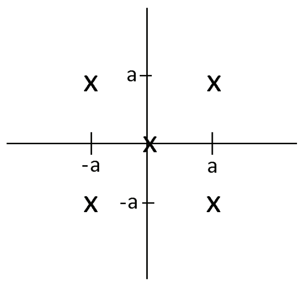


Constellation 1

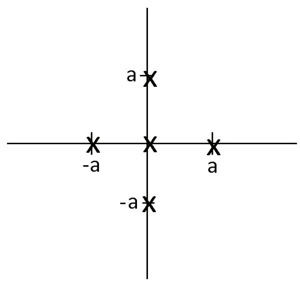


Constellation 2

(c) (4 pts)

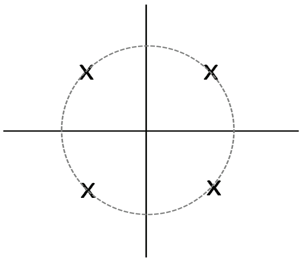


Constellation 1

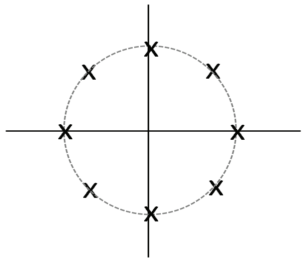


Constellation 2

(d) (8 pts)



Constellation 1



Constellation 2

3. **Punctured Convolutional Codes** (35 pts)
High-rate convolutional codes can be derived from lower rate convolutional codes through the process of *puncturing*. Puncturing refers to deleting some of the coded bits output by the encoder. By deleting (i.e., puncturing) some of the bits, the total number of coded bits is decreased and therefore the rate of the code is increased. (The pattern with which bits are deleted, i.e., the puncturing pattern, is fixed and is known to both the encoder and decoder.)

   (a) We begin with the $R = 1/2$ binary convolutional code with memory 2 with the following outputs:

$$\begin{aligned} u[k] + u[k-1] + u[k-2] \\ u[k] + u[k-2] \end{aligned}$$

   This is the first code we studied in class, and in octal notation this is the [7,5] code (nonsystematic and nonrecursive). We will puncture this code to create a $R = 2/3$ code.

   Considering the following puncturing pattern: the *second* coded bit corresponding to the *first* information bit is punctured, the *second* coded bit corresponding to the *third* information bit is punctured, the *second* coded bit corresponding to the *fifth* information bit is punctured, and so on. The general pattern is that the the *second* coded bit corresponding to every other information bit is punctured.

   For example, consider the information bit sequence 1100: the un-punctured output sequence from the normal $[7,5]$ encoder is 11010111, whereas the punctured sequence is $1\cancel{1}010\cancel{1}11 \;\rightarrow$ 101011. With this pattern we have 3 coded bits for every 2 information bits, and thus the code has rate $2/3$.

   Compute d_{free} for this punctured code, and indicate the corresponding error event on the provided trellis. (10 pts)

   Note: You are provided with an extended trellis diagram for the *un-punctured* $[7,5]$ code - you will need to modify some of the transition labels to account for puncturing.

(b) We now consider a *different* puncturing pattern: the *first* coded bit corresponding to the *first* information bit is punctured, the *first* coded bit corresponding to the *third* information bit is punctured, and so on. The general pattern is that the the *first* coded bit corresponding to every other information bit is punctured.

For example, consider the information bit sequence 1100: the un-punctured output sequence from the normal $[7, 5]$ encoder is 11010111, whereas the punctured sequence is $\cancel{1}101\cancel{0}111 \rightarrow 101111$. This is also a rate 2/3 code.

i. Verify that this punctured code has the same d_{free} as in the previous part, and indicate the corresponding error event on the provided trellis. (5 pts)

ii. Verify that this punctured code has an *infinite* number of error events with output weight $1 + d_{\text{free}}$, and indicate the corresponding error events on the provided trellis. (5 pts)

(c) The rate of a parallel concatenated turbo code can be increased by puncturing the outputs of encoder 1 and encoder 2. In class we considered a $R = 1/3$ turbo code that was composed of two $R = 1/2$ systematic and recursive convolutional encoders (in parallel). To increase the rate, we puncture every other parity bit generated by encoder 1, and we puncture every other parity bit generated by encoder 2. We do not puncture the information (i.e. systematic) bits.

i. What is the rate of the resulting turbo code? (3 pts)

ii. Explain how you can decode this *punctured* turbo code using a turbo decoder designed for the *un-punctured* $R = 1/3$ code. You should not make any changes to the un-punctured decoder. (12 pts)

Note: In practice, a turbo decoder designed for the un-punctured code is used to decode un-punctured and punctured turbo codes. This provides a substantial savings in hardware since the same circuit is used to decode multiple codes.

4. **Equalization** (35 pts)
Consider a system where the impulse response of the TX filter and of the channel are:

$$g_{\mathrm{TX}}(t) = \begin{cases} 1 & 0 \le t \le 2 \\ 0 & \text{else} \end{cases} \qquad\qquad g_C(t) = \delta(t) + \delta(t - 1.5)$$

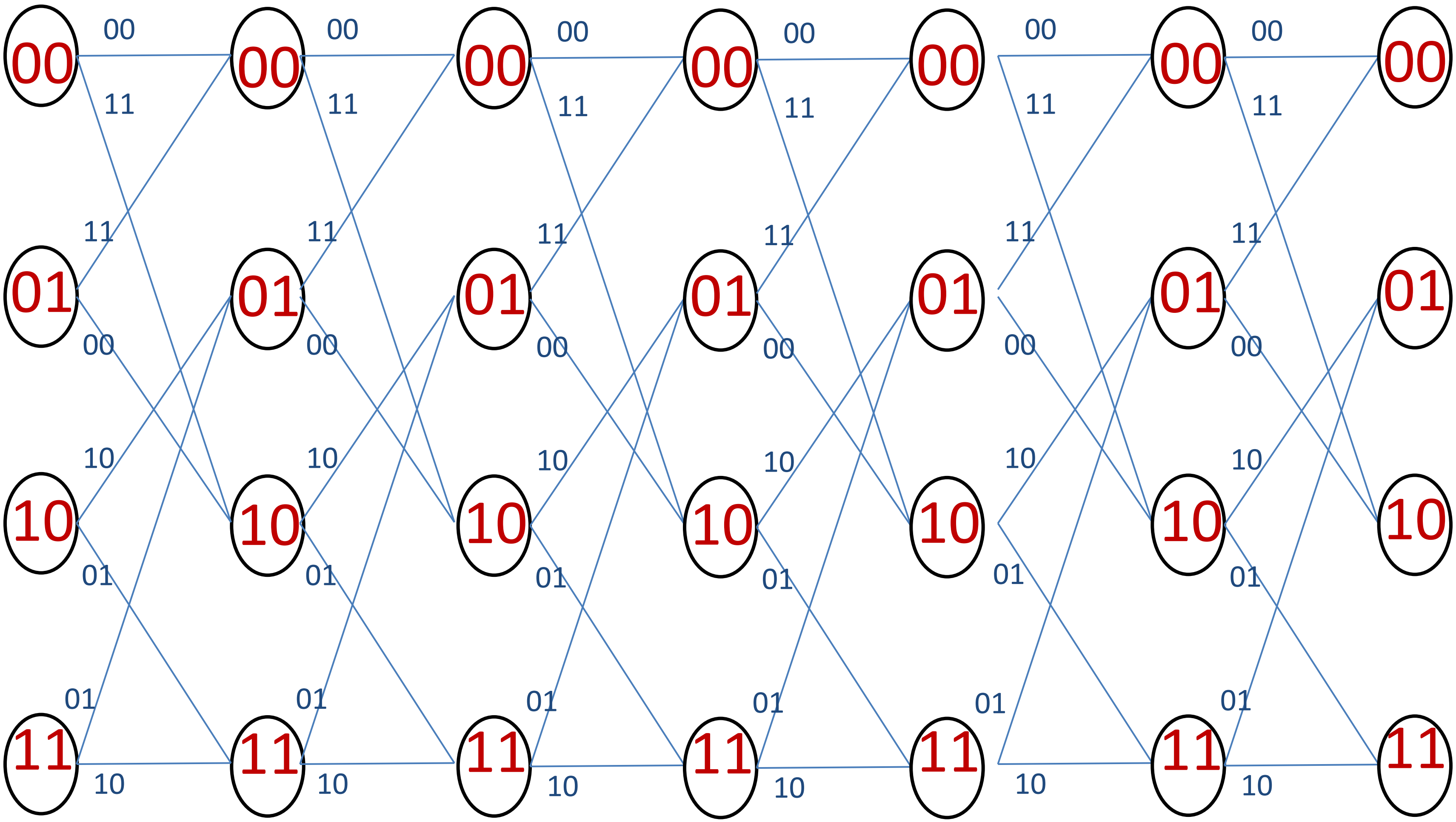
The symbol period is $T = 2$, i.e., one symbol every two time units.

(a) Compute $p(t)$ and $h[n]$. (5 pts)

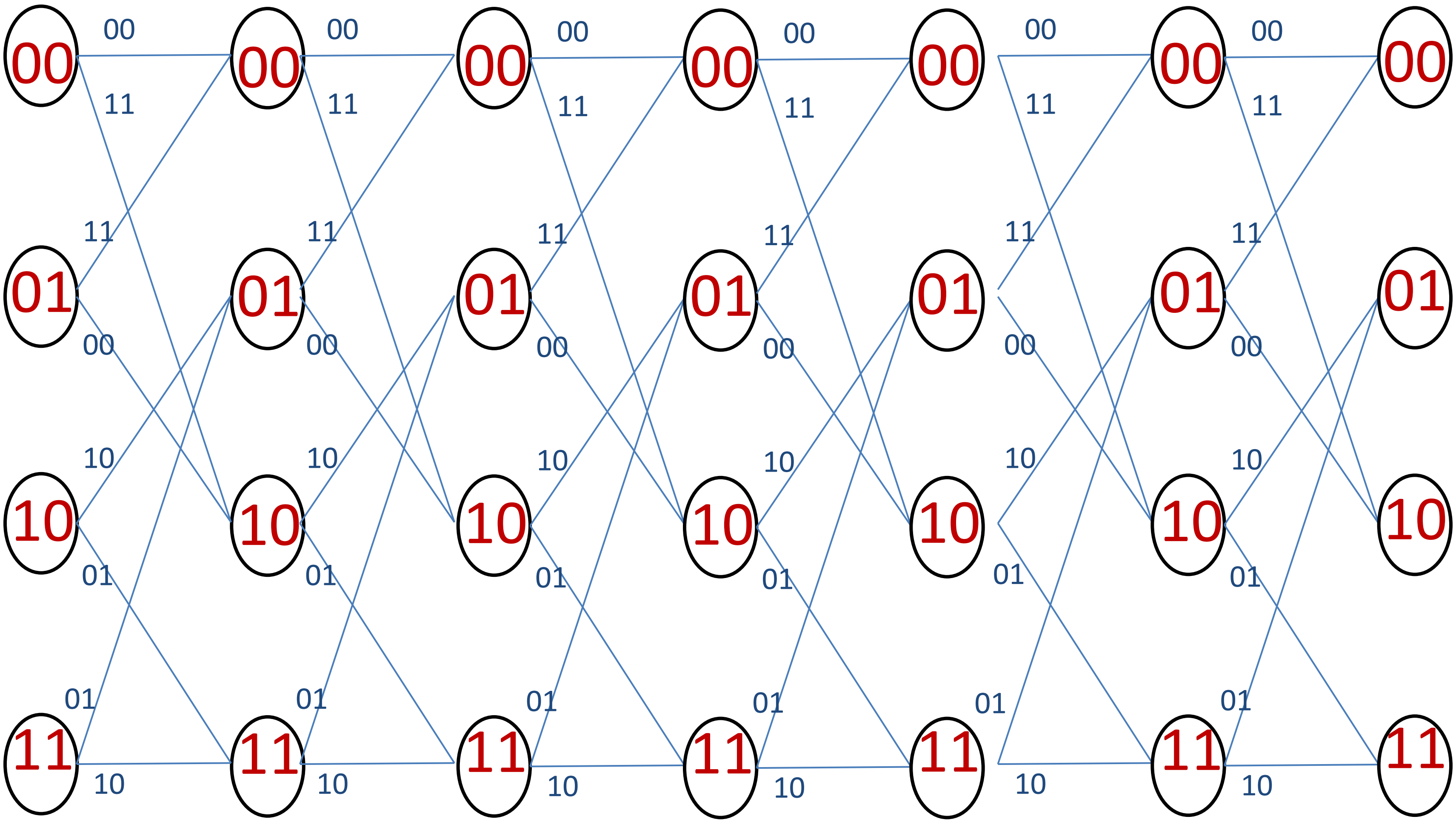
(b) If MLSE is to be performed (assuming BPSK), how many states would be required in the Viterbi implementation? (5 pts)

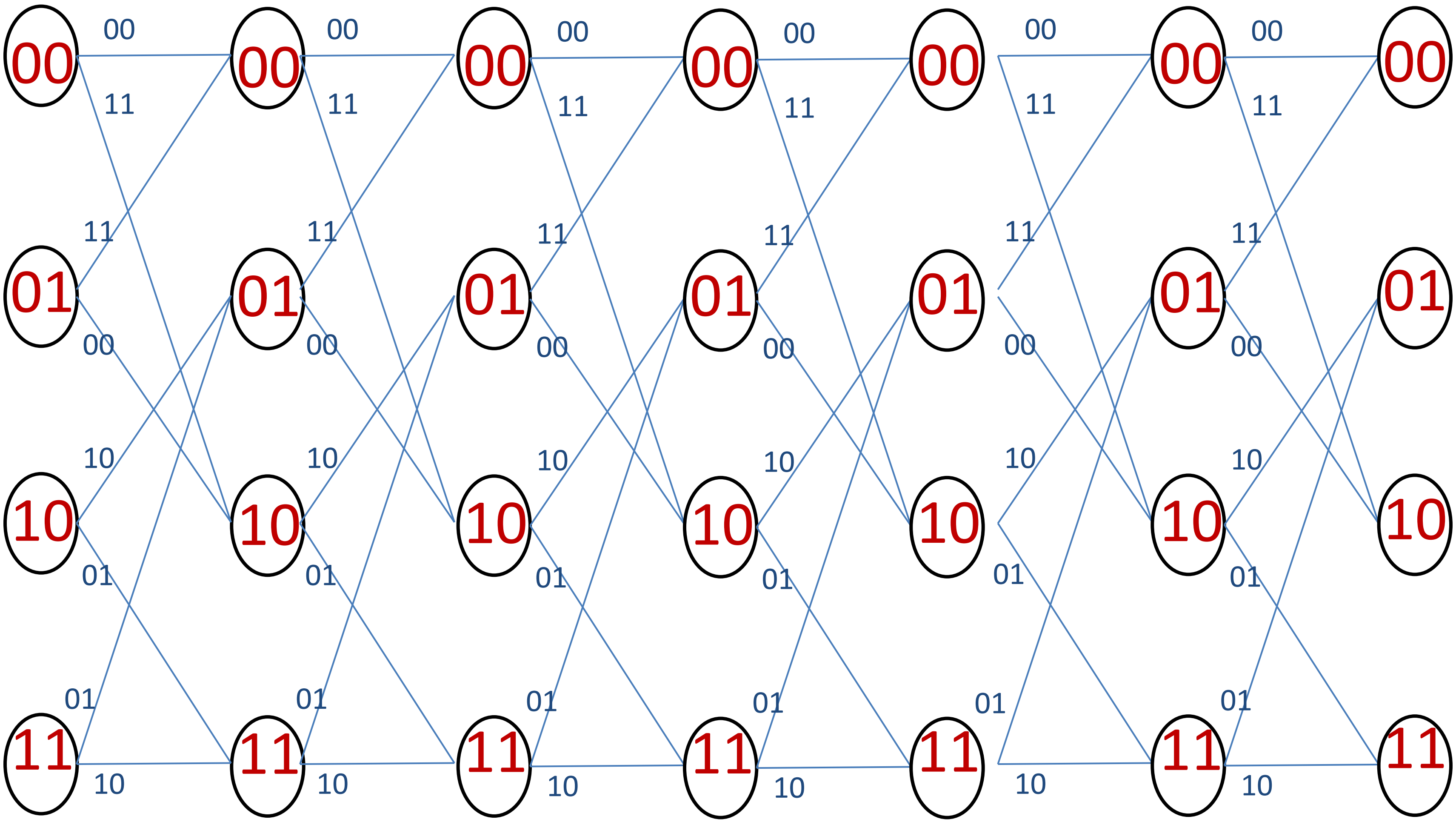
(c) Assume white Gaussian noise with PSD σ^2 and that BPSK with ± 1 is used. We are interested in the MMSE equalizer (on the standard matched filter outputs) of length 3. Compute $\mathbf{U}$, the matrix that maps from bits to the 3 received symbols, and $\mathbf{C}_w$, the noise covariance matrix (for 3 received symbols), and write out the equation for the MMSE received filter. (15 pts)

(d) Find an RX filter and an appropriate sampling period such that one of the RX filter samples is interference-free (i.e., it only contains the contribution of a single transmitted bit/symbol and noise). (10 pts)



00
00
00
00
00
00
00
00
00
00
00
00
00
11
11
11
11
11
11
11
11
11
11
11
11
01
01
01
01
01
01
01
00
00
00
00
00
00
10
10
10
10
10
10
10
10
10
10
10
10
10
01
01
01
01
01
01
01
01
01
01
01
01
11
11
11
11
11
11
11
10
10
10
10
10
10


00 00 00 00 00 00 00
00 00 00 00 00 00 00
11 11 11 11 11 11
11 11 11 11 11 11
01 01 01 01 01 01 01
00 00 00 00 00 00
10 10 10 10 10 10
10 10 10 10 10 10 10
01 01 01 01 01 01
01 01 01 01 01 01
11 11 11 11 11 11 11
10 10 10 10 10 10


00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
11 11 11 11 11 11 11
11 11 11 11 11 11 11
01 01 01 01 01 01 01
00 00 00 00 00 00 00
10 10 10 10 10 10 10
10 10 10 10 10 10 10
01 01 01 01 01 01 01
01 01 01 01 01 01 01
11 11 11 11 11 11 11
10 10 10 10 10 10 10