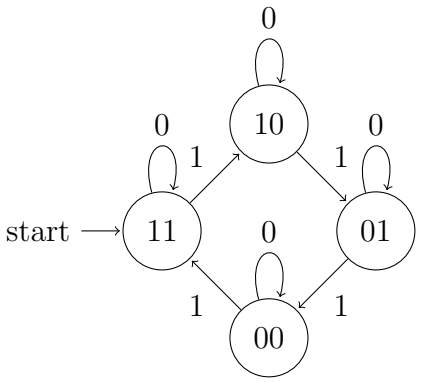
1. The state diagram is like below:



The state table is like the following:

x	Current State	Next State
0	00	00
0	01	01
0	10	10
0	11	11
1	00	11
1	01	00
1	10	01
1	11	10

Current State / x	00	01	11	10
0	0	0	1	1
1	1	0	1	0

Table 1: Karnugh Map for D_1

Current State / x	00	01	11	10
0	0	1	1	0
1	1	0	0	1

Table 2: Karnugh Map for D_2

The circuit is in the attached files.

2. (a) 1- Capacity of the memory. 2- Speed of the memory. 3- Cost of the memory.

   (b) We use memory hierarchy to have the maximum capacity and speed and minimum cost. The method is that you use memories with higher speed in the place in which they are closer to the CPU and memories with less speed but more capacity further from the CPU. To have maximum speed and capacity and minimum cost.

   (c) Yes, when the time of retrieving data from the intermediate memories is more than the time of retrieving data from the main memory.

   (d) Static memories are more expensive and have high speed because they don't need refreshing and less dense and they use less power. But Dynamic memories are chipper but they need refresher and so they are slower than static memories and they are denser than static memories and they use more power. - Using static memories is easier. The picture is in the attached files.

3. $T_{access} = t_1 + (1 - h_1)(t_2 + (1 - h_2)(t_3 + (1 - h_3)t))) = 3 * 10^{-9} + 0.08 * (20 * 10^{-9} + 0.25 * (350 * 10^{-9} + 0.65 * 12 * 10^{-3})) = 1.56 * 10^{-4} s$

   (a) $T_{access} = t_1 + (1 - h_1)(t_2 + (1 - h_2)(t_3 + (1 - h_3)t))) = 3 * 10^{-9} + 0.08 * (20 * 10^{-9} + 0.30 * (350 * 10^{-9} + 0.65 * 12 * 10^{-3})) = 1.87213 * 10^{-4}$

   It is 0.83 times faster than the first system. This means this system is actually slower.

   (b) $T_{access} = t_1 + (1 - h_1)(t_2 + (1 - h_2)(t_3 + (1 - h_3)(t_4 + 1 - h_4(t_5)))) = 3 * 10^{-9} + 0.08 * (8 * 10^{-9} + 0.15 * (20 * 10^{-9} + 0.3 * (10^{-9} * 350 + 0.65 * 10^{-3} * 12))) = 2.81 * 10^{-5}$

   It is 5.551 times faster than the first system.

4. (a) height = bitline = 8B = $8 * 8 = 64$ bits.
   width = wordline = 64 * 1024 = 65536 bits.
   minimum accessible data = 8B.
   volume = size = $64 * 65536 = 4194304$ bits.
   number of bits for addressing = $\log_2 65536 = 16.0$

   (b) height = wordline = 1B = $1 * 8 = 8$ bits.
   width = bitline = 16 * 1024 = 16384 bits.
   minimum accessible data = 1B.

volume = size $= 8 * 16384 = 131072$ bits.
number of bits for addressing $= \log_2 131072 = 17.0$

(c) height = wordline = 4B $= 4 * 8 = 32$ bits.
width = bitline $= 2^{13} = 8192$ bits.
minimum accessible data = 4B.
volume = size $= 32 * 8192 = 262144$ bits.
number of bits for addressing = 13 bits