4. (i) The global DB needs to store the current state of the game. This involves the state of the grid, plus an indicator of whose turn it is. The state of the grid can be represented as a 9-D vector, with say '1' used for X, '2' used for O, and '0' used for an empty square. So for the given example grid state, the vector representation would be: {0, 2, 1, 2, 1, 1, 2, 0, 0}. In addition we can have a variable called 'NEXT', NEXT = 1 could denote that the player with symbol 1 (X) goes next, and NEXT = 2 for the other player.

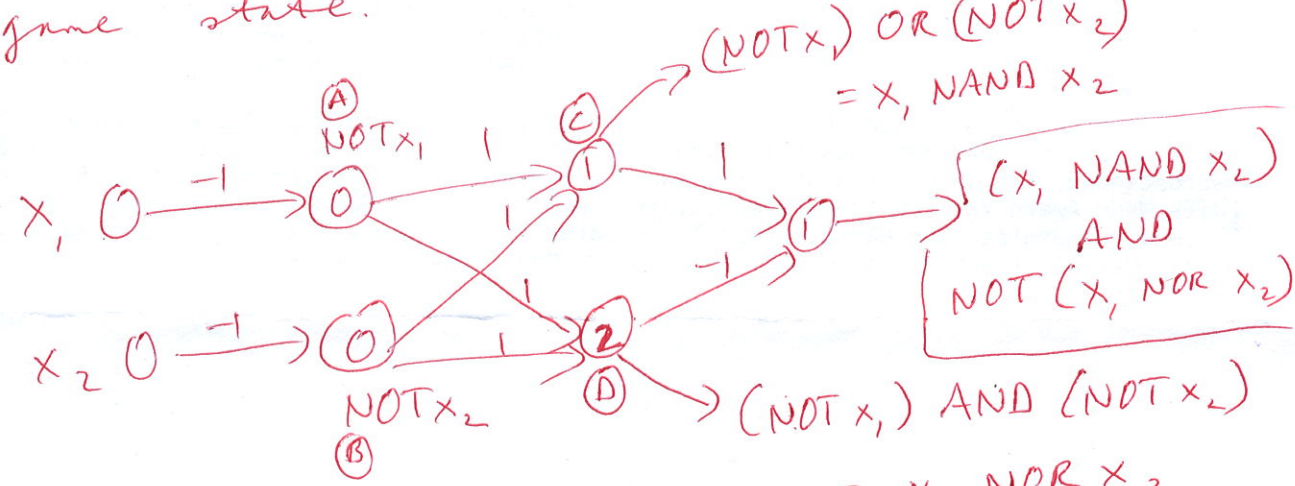
(ii) Here are two possible rules (positions on grid numbered as in above example from 1 to 9):

wins game for '1' → If position 1 = '1' and position 4 = '1' and position 7 = '0' and NEXT = 1, then set position 7 = '1' and NEXT = 2

prevents possible win for '1' → If position 1 = '1' and position 4 = '1' and position 7 = '0' and NEXT = 2, then set position 7 = '2' and NEXT = 1

(iii) The control structure will be the strategy used by the players to pick one of the applicable production rules at each turn. For example, each rule might have an associated probability of it winning you the game, based on forward reasoning or past experience. Then your strategy might be to pick the rule with the highest win probability, which is applicable to the current game state.

5.



Output = C AND NOT D	x_1	x_2	A: NOT x_1	B: NOT x_2	C = A OR B = x_1 NAND x_2	D = A AND B = x_1 NOR x_2
0	0	0	1	1	1	0
1	0	1	1	0	1	0
1	1	0	0	1	1	0
0	1	1	0	0	0	0

→ So the output is x_1 XOR x_2.