Question One: Convert the following NFA to DFA.

\[ \text{ANSWER:} \]
Question 2: Convert the following NFA into the equivalent DFA.

What language does this NFA accept?
The top half of the NFA is the $x \equiv 0 \mod 3$ machine from the homework. The bottom half of the NFA is a $x \equiv 0 \mod 4$ machine. The NFA is the union of these two machines. Therefore $L(M) =$ binary strings $x$ such that $x \equiv 3$ or $x \equiv 0 \mod 4$. 
**Question 3:** (a) Convert the following NFA with $\land$ transitions into an NFA without $\land$ transitions.

(b) Convert the NFA into a DFA.
answer 3:

(a)
Question 4:

Let $L_2$ be the set of strings over $\{0,1\}^*$ that contain exactly 2 0's or 2 1's. Examples:

- 0011 is in $L_2$
- 110011111 is in $L_2$
- 1000101 is not in $L_2$

Construct a NFA that accepts the language described in problem 2. What does each state represent?

Suppose our language was the set of strings containing exactly 3 0's or 3 1's. How many states would our NFA need?

**Answer to Question 4**

![NFA Diagram]

The state names indicate the meaning of each state. State $Qx0$ is the state we are in after seeing an arbitrary number of 0's and zero 1's, state $Q1x$ is the state we are in after seeing an arbitrary number of 1's and one 0, etc.

We would only need 9 states to recognize a language consisting of strings with exactly 3 0's or 3 1's.