Consider the operation of a very “simplified” microwave oven. The oven works according to the following specification. The user enters the number of seconds to cook. S/he then presses the on button. The oven begins cooking only if the oven door is closed. If it is, the control decrements the number of seconds left to cook once per second. If the door is ever opened while cooking, the oven immediately stops cooking (and decrementing the number of seconds left to cook) until the door is closed again.

We will assume that the number of seconds to cook is provided externally, and as far as the oven control is concerned, it is simply an input of a binary magnitude number (Cook_time) that is counted down to zero. There is a Decrement_count output that will subtract one from the cook_time once per clock cycle while this signal is asserted. A Cook-on output turns on the cooking element when it is true; when it is false, the element is turned off, and cooking stops. The oven has an On input that is true when the on button is pushed, and a separate input off when the Off button is pressed. Another input, Door_open, is true whenever the oven door is open, false otherwise. For simplicity’s sake, we will assume that the controller runs at the speed of one clock cycle per second.

A Reset input puts the state machine in a well-known starting state that we call off. Any time the off button is pressed, we go to the off state. There are two more states, cook and idle. Pressing the on button turns on the oven on, and places it in the cook state. If cooking, and the door is opened, then go to the idle state until the door is closed again.

A block diagram for the controller is as follows:

Complete the MEALY MACHINE state diagram for this controller on the back of this sheet. Label each transition you add to the diagram with the input conditions and the outputs that are to be asserted (by default, if an output is not asserted on a transition, we will assume it to be unasserted for that input and state).