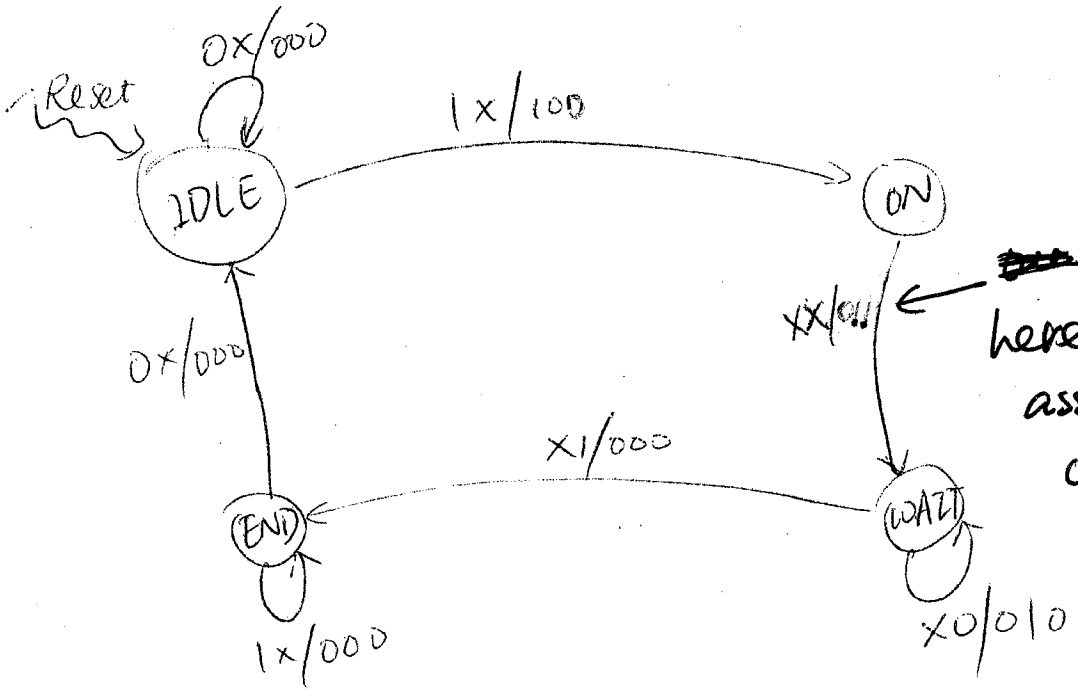


HW6 solution (modified)

3(a)

input: start echo

output: clearCount, enableCount, ping



~~emit ping~~ here so that it's asserted for one clock cycle

wait in this state until start signal is off.

ignore the start signal once the start button is pressed. turn off the ping signal after 1 cycle.

Note: The "END" state is used to wait till the start signal is turned off. It's OK ~~that~~ if your design doesn't include this state.

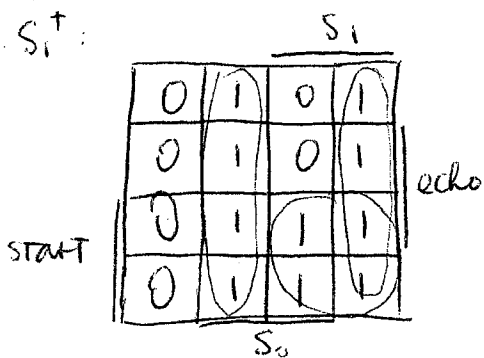
(b)

start	echo	$S_1$	$S_0$	$S_1^+$	$S_0^+$	clearCount	EnableCount	ping	
0	0	IDLE	00	IDLE	00	0	0	0	
0	1	IDLE	00	IDLE	00	0	0	0	
1	0	IDLE	00	ON	01	1	0	0	
1	1	IDLE	00	ON	01	1	0	0	
0	0	ON	01	WAIT	10	0	1	1	
0	1	ON	01	}	}	0	1	1	
1	0	ON	01			0	0	1	1
1	1	ON	01			0	0	1	1
0	0	WAIT	10	WAIT	10	0	1	0	
0	1	WAIT	10	END	11	0	0	0	
1	0	WAIT	10	WAIT	10	0	1	0	
1	1	WAIT	10	END	11	0	0	0	
0	0	END	11	IDLE	00	0	0	0	
0	1	END	11	IDLE	00	0	0	0	
1	0	END	11	END	11	0	0	0	
1	1	END	11	END	11	0	0	0	

(c) state assignment:

IDLE = 00      ON = 01      WAIT = 10      END = 11

(d)  $S_1^+$ :



$$S_1^+ = S_1 \bar{S}_0 + \bar{S}_1 S_0 + S_1 \text{start}$$

$S_0^+$

	$S_1$				
	0	0	0	0	
	0	0	0	1	echo
start	1	0	1	1	
	1	0	1	0	
	$S_0$				

$$S_0^+ = S_1 \bar{S}_0 \text{ echo} + S_1 S_0 \text{ start} + \bar{S}_1 \bar{S}_0 \text{ start}$$

clearCount:

	$S_1$				
	0	0	0	0	
	0	0	0	0	echo
start	1	0	0	0	
	1	0	0	0	
	$S_0$				

$$\text{clearCount} = \bar{S}_1 \bar{S}_0 \text{ start}$$

enableCount:

	$S_1$				
	0	1	0	1	
	0	1	0	0	echo
start	0	1	0	0	
	0	1	0	1	
	$S_0$				

$$\text{enableCount} = \bar{S}_1 S_0 + S_1 \bar{S}_0 \text{ echo}$$

ping:

	$S_1$				
	0	1	0	0	
	0	1	0	0	echo
start	0	1	0	0	
	0	1	0	0	
	$S_0$				

$$\text{ping} = \bar{S}_1 S_0$$

(e)

