Design Project 3

CMPEN 270 Spring 2022

Dohyoung Ko

Electrical Engineering and Computer Science
College of Engineering
Pennsylvania State University

4/22/2022

Design of a switching power converter Part I

(a) Duty Cycle Binary Representation Table

Q_1Q_0	Duty Cycle (d)
00	0%
01	25%
10	50%
11	75%

(b) Table

Input Condition	Voltage Control Inputs	Action Taken
$V_{boost} < 1.8V$	$R = 0$	Increase to maximum duty cycle (Asynchronous reset)
$1.8V \leq V_{boost} < 1.9V$	$R = 1$, $X = 0$, $Y = 1$	Increase duty cycle to its next highest value
$1.9V \leq V_{boost} < 2.1V$	$R = 1$, $X = 0$, $Y = 0$	None
$V_{boost} \geq 2.1V$	$R = 1$, $X = 1$, $Y = 0$	Reduce duty cycle to next lowest value

(c) State Transition Table

Assuming that a command to decrease d when at 0% produces a 0% next state, and increasing d past 75% results in 75% for the next state, Table 1 is the state transition table. It is written in a grey-code order.

In this state transition table, current state is represented as Q_1Q_0. Next state is $Q_1^*Q_0^*$. Input I, which is voltage control inputs X and Y are together represented as two bit binary number $I = XY$. For example, $XY = 01$ signals that the input condition is $1.8V \leq V_{boost} < 1.9V$ and the power converter should increase duty cycle to its next highest value.

Current State	Next State ($Q_1^*Q_0^*$)			
(Q_1Q_0)	$XY = 00$	$XY = 01$	$XY = 11$	$XY = 10$
00	00	01	dd	00
01	01	10	dd	00
11	11	11	dd	10
10	10	11	dd	01

Table 1: State Transition Table of Part I

(d) State Machine Diagram

Figure 1 is the state machine diagram corresponding to Table 1. Input XY is drawn as a tuple (X, Y) on each arrow.

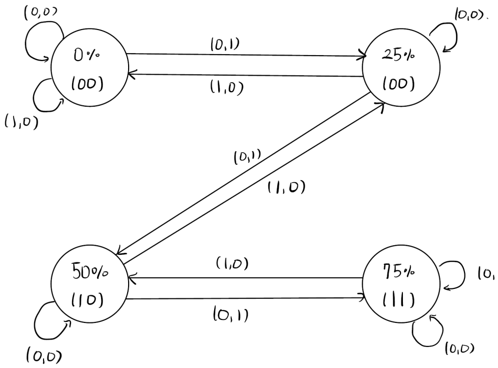
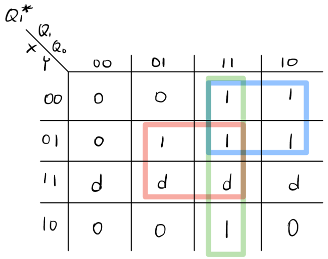


Figure 1: State Machine Diagram of Part I

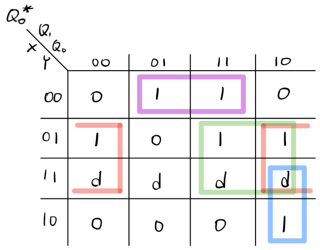
(e) Logical Functional Expressions and Logic Circuit

Figure 2 is the K-map of state variable Q_1^*. Figure 3 shows the K-map of state variable Q_0^*.



Q_1^* XY \ Q_1Q_0	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	d	d	d	d
10	0	0	1	0

Figure 2: K-map of Q_1^*



Q_0^* XY \ Q_1Q_0	00	01	11	10
00	0	1	1	0
01	1	0	1	1
11	d	d	d	d
10	0	0	0	1

Figure 3: K-map of Q_0^*

From above K-maps, we obtain the equations for Q_1^* and Q_0^*:

$$Q_1^* = Q_0Y + Q_1\overline{X} + Q_1Q_0 \tag{1}$$

$$Q_0^* = Y\overline{Q_0} + YQ_1 + \overline{X}Q_0\overline{Y} + XQ_1\overline{Q_0} \tag{2}$$

Using both equations, we can obtain the circuit by using two positive-edge triggered D Flip Flops for state variables Q_0 and Q_1. Figure 4 is the circuit diagram for part 1.

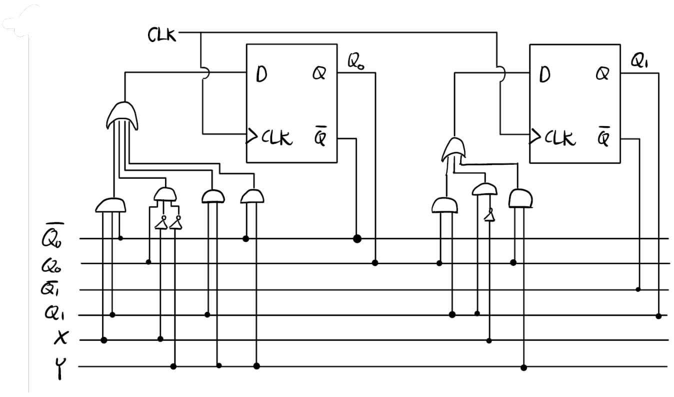


Figure 4: Circuit Diagram

(f) Design Restrictions

If we did not incorporate the restrictions about these specific cases, then we will treat them as don't cares in our state transition table, corresponding K-map, and the equations. Having a don't care in truth table may result in a different circuit design which do not behave equally. This is okay if the input corresponding to those don't cares is invalid and never used while the user operates the circuit. In this example, input $I = XY = 11$ is never used, so it's okay to put don't care signs. However, aforementioned situations from part (c) can possibly happen, meaning that we should be able to control these states. Without restrictions, we don't have a full control on these cases. This is a great obstacle where we should be able to expect every behavior of a circuit. Therefore, it was necessary to incorporate the restrictions from part (c).

Design of a switching power converter Part II

(a) State Transition Table

Current state is represented as C_1C_0. Next state and output are marked as $(C_1^*C_0^*, S)$. Input, which is a duty cycle is a 2 bit binary number $d = d_1d_0$. For example, $d = d_1d_0 = 01$ means 25% duty cycle.

Current State	(Next State, Output) $(C_1^*C_0^*, S)$			
(C_1C_0)	$d_1d_0 = 00$	$d_1d_0 = 01$	$d_1d_0 = 10$	$d_1d_0 = 11$
00	(01, 0)	(01, 1)	(01, 1)	(01, 1)
01	(10, 0)	(10, 0)	(10, 1)	(10, 1)
10	(11, 0)	(11, 0)	(11, 0)	(11, 1)
11	(00, 0)	(00, 0)	(00, 0)	(00, 0)

Table 2: State Transition Table of Part II

(b) State Machine Diagram

Figure 5 is the state machine diagram corresponding to Table 2. Input d_1d_0 and output S are interpreted as a tuple (d_1, d_0) / S. For example, (0, 1), (1, 1), (1, 0) / 1 with a single arrow means for inputs (0, 1), (1, 1), and (1, 0), the machine outputs 1 and all these go to the same next state.

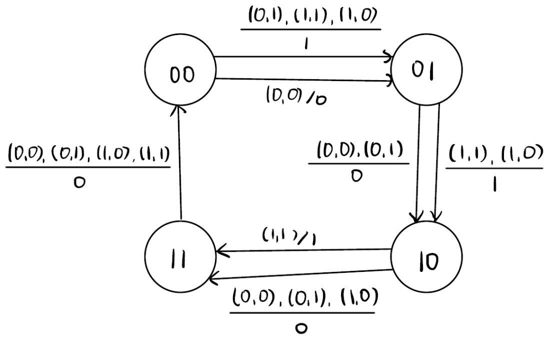
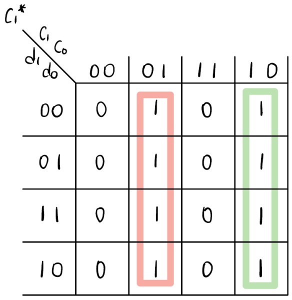


Figure 5: State Machine Diagram of Part II

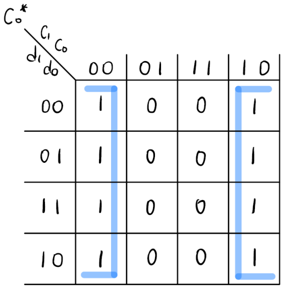
(c) Logical Functional Expressions and Logic Circuit

Figure 6 is the K-map of state variable C_1^*. Figure 7 is the K-map of state variable C_0^*.



C_1^* $C_1 C_0$ $d_1 d_0$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

Figure 6: K-map of C_1^*



C_0^* $C_1 C_0$ $d_1 d_0$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

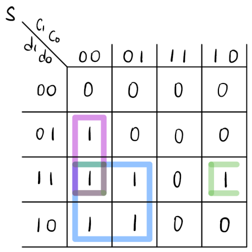
Figure 7: K-map of C_0^*

From those K-maps, we obtained the equation for C_1^* and C_0^*.

$$C_1^* = \overline{C_1}C_0 + C_1\overline{C_0} = C_1 \oplus C_0 \tag{3}$$

$$C_0^* = \overline{C_0} \tag{4}$$

Finally, Figure 8 is the K-map of the output S.



S C_1C_0 d_1d_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

Figure 8: K-map of S

From the K-map, we obtained the equation for S:

$$S = \overline{C_0} d_0 \overline{C_1} + \overline{C_0} d_1 d_0 + \overline{C_1} d_1 \tag{5}$$

Using above three equations, we can obtain the logic circuit by using two positive-edge triggered D Flip Flops for state variables C_0 and C_1. Figure 9 is the circuit diagram for part 2.

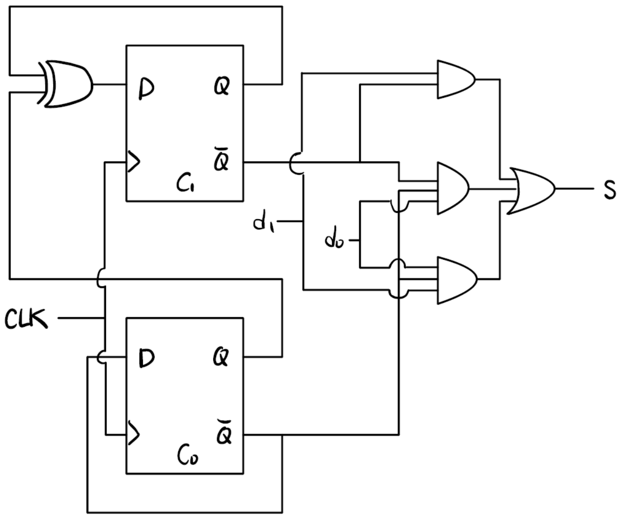


Figure 9: Circuit Diagram

(d) Timing Diagrams

1. 25% Duty Cycle

Figure 10 shows the timing diagram for S for duty cycle $d = d_1 d_0 = 01 = 25\%$.

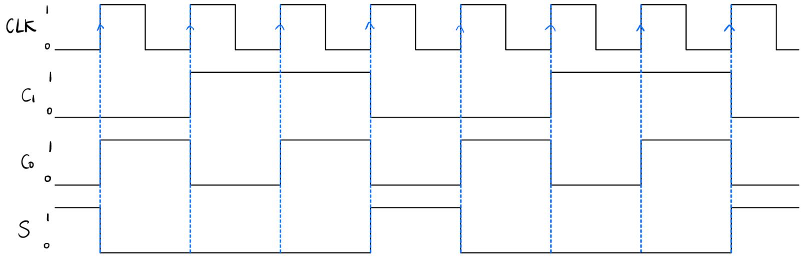


Figure 10: 25% Duty Cycle Timing Diagram for S

We observe that the $1 : 0$ ratio of the output signal S is $1 : 3$, which is equivalent to the duty cycle of $\frac{1}{4} = 25\%$.

2. 75% Duty Cycle

Figure 11 shows the timing diagram for S for duty cycle $d = d_1 d_0 = 11 = 75\%$.

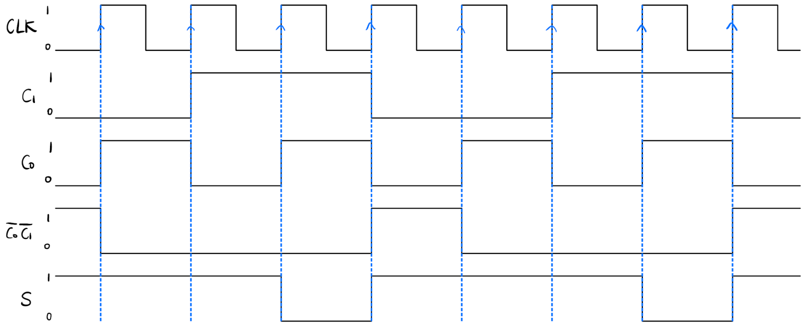


Figure 11: 75% Duty Cycle Timing Diagram for S

We observe that the $1 : 0$ ratio of the output signal S is $3 : 1$, which is equivalent to the duty cycle of $\frac{3}{4} = 75\%$.

(e) Maximum Switching Frequency of the Converter

Given the clock frequency of 1MHz, the converter's maximum switching frequency is 0.25MHz. The converter circuit is based on a 2-bit modulo counter. Therefore, a single switching cycle of S depends on 4 clock cycles. Since the clock frequency is 1MHz, the counter's frequency can be at max of $\frac{1}{4} = 0.25$MHz.

Since flip flop is either positive or negative edge-triggered, each state variable C_1 and C_0 can switch at most $\frac{1}{2}$ the rate the clock signal changes. Therefore, after an entire clock cycle, state variables C_1 and C_0 can complete at most $\frac{1}{2}$ of its cycle. Since converter output S depends on the state of both C_1 and C_0, the converter can switch at most $\frac{1}{4}$ of its cycle per clock cycle. This is equivalent to say that the maximum switching frequency of the converter is $1 \times \frac{1}{4} = 0.25$ MHz.

In general, given the clock frequency of f_{clk} and n number of bits, the counter's maximum clock frequency f_s is expressed as

$$f_s = \frac{f_{clk}}{2^n} \tag{6}$$

(f) Design Tradeoff

To design a circuit with a highly precise duty cycle, we need a counter with more bits, that is, n increases in equation (6). As a result, f_s decreases, meaning that the maximum clock frequency decreases, and the circuit becomes slower. Along with the speed issue, the circuit also consumes more energy and is expensive since a lot more gates resources are required.

Therefore, when the cost and resources are more important factors than precision and speed, it would be more efficient to have less precise duty cycles.

There is one example. Consider a desktop computer and a graphing calculator. We do not expect a desktop-level computing speed from a graphing calculator. Also, a calculator should prioritize its power management over the huge calculation speed or precision because it does not have a consistent power source; it runs on batteries. In this case, it would be wise to use less precise duty cycles for a graphing calculator.

Design of a switching power converter Part III

(a) DEB Pin Information

In my DEB-1002, I used pin IN1 and IN0 for input d_1 and d_0, respectively. For example, if I set (IN1, IN0) to (1, 0), the circuit will produce a corresponding waveform of 50% duty cycle.
Since DEB-1002 has a builtin clock signal, I set the CLOCK_ENABLE to 1 and connected CLOCK pin to the inputs of my two positive edge triggered D Flip

Flops(SN74HC74N). Finally, I used the pin OUT0 to demonstrate the working waveform.

(b) Circuit Implementation

From the logical expressions obtained above, I made some changes to the final equation in order to reduce the complexity and required amount of gate resources.

$$S = \overline{C_0}d_0\overline{C_1} + \overline{C_0}d_1d_0 + \overline{C_1}d_1 = \overline{C_0}d_0(d_1 + \overline{C_1}) + \overline{C_1}d_1 \tag{7}$$

In total, I used 1 XOR, 2 OR, 3 AND gates. Also, I used 2 positive edge triggered D Flip Flops to store state variables. Figure 12 shows the circuit implementation in DEB-1002.

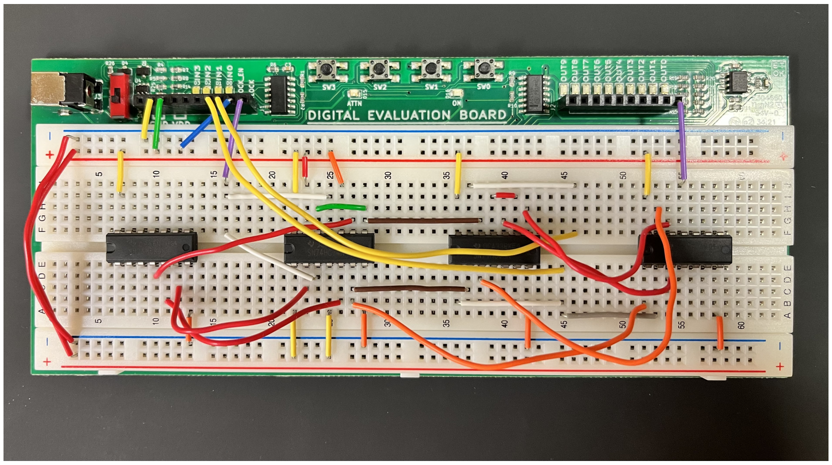


Figure 12: Final Circuit Implementation

(c) Link to the Video Demo

Here is the link to the video demo: https://youtu.be/EXsVrjeMC9M