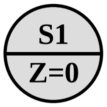
Dr. Bassem Al-Halabi, S&E362 November 26, 2000

A Comprehensive Example on Sequential Circuit Design

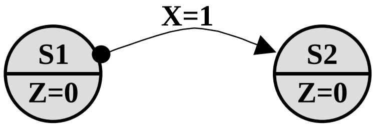
State Graph Design (Moore Style)

1) The father is concerned about his son's performance at the school and wants to make up an incentive for him to improve his grades. He told him every time you get A's in three consecutive assignments, I buy you a gift. The son started working hard and the father started designing a circuit, which helps him remember how close his son is to getting the gift. The father, taking Logic Design at FAU, realized that he is about to design a *logic circuit* because it involves some logical relation between a certain condition (the A-grade) or *input* and a certain result (signal to buy a gift) or *output*. He further realized that the circuit has to remember the steps of the status of his son getting closer to the gift, so he decided to use some memory elements or *flip-flops*. Finally he realized that those steps of status, or simply *states*, have to be in a certain order or sequence, and that made him conclude that he is designing a *sequential circuit*.

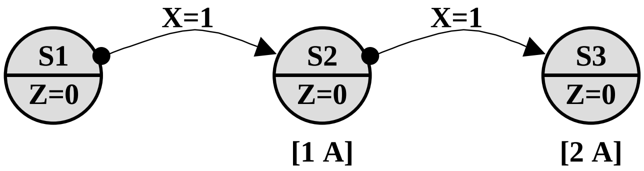
To start, he assumed he is looking at an initial point of time which corresponds to the status state where the son has no A-grade yet, and he called it state **S1**. At that state **S1,** the output, which he called **Z**, of the circuit is **Z=0** indicating no gift yet. To help remember the sequence of events in the design, the father used a special form of flowchart, which is called *state graph* as shown below. The circle is a state, the upper label is the state number and the lower label is the value of output **Z**.



Now, when the son makes his first A-grade inputted to the circuit by an input signal **X=1**, the state graph has to move to another state, say **S2**, to remember that the son has received his first A. (**X=0** for grades other than A). The output is still **Z=0**, indicating to the father that no gifts yet.



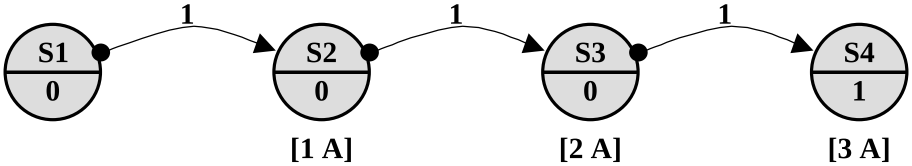
When the son receives the second A-grade, then the state graph will move to another state, say **S3**, to remember that the son has received 2 A-grades so far. For the father to remember which state means what, he indicated next to each state the number of A-grades received so far as shown below:



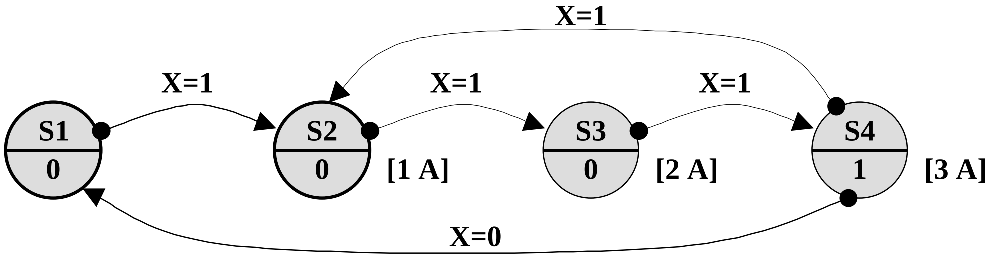
In the same fashion, when the son receive the third A-grade, a new state is drawn as shown below. The difference however, is that the output must now be **Z=1** to signal the father to buy the gift. To simplify the graph, phrases **X=** on the arrows and **Z=** in the states are omitted.

Dr. Bassem Al-Halabi, S&E362 November 26, 2000

A Comprehensive Example on Sequential Circuit Design



These are all possible states that must be remembered and if the son starts with **S1** and receives 3 consecutive A-grades as indicated by **X=1**, he will reach status state **S4** and get the gift as indicated by output **Z=1**. Now after the third A-grade, if the son receives a bad grade, then he will return to state **S1** to start all over again. But if he receives another A-grade, he will return to state **S2**, which remembers having received the first A-grade.



The question now is if the son is in state **S1**, **S2** or **S3** and receives a bad grade (**X=0**), where would he go? The answer relies in the condition that the three A-grades must be consecutive, and therefore, if the son interrupts the desired sequence (**X=1** three consecutive times), he must start all over again by returning to **S1**. This will complete the state graph as follows:

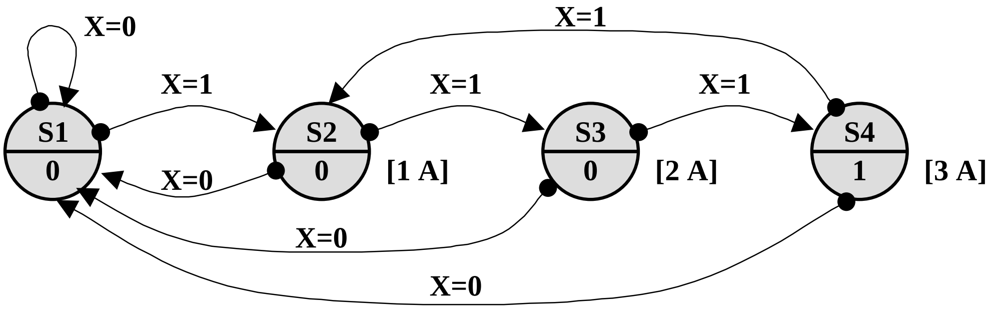


Figure 1. Output Z=1 if and only if X=1 three consecutive times, else start all over again

The state graph is considered complete when each and every state has exactly two outgoing arrows, one for **X=1** and one for **X=0**. The father has finally built the circuit of Figure 1 (the subject of later sections) and used at home to monitor his son's performance.

2) Much time has passed and the son never got a gift, and therefore started losing interest. He worked so hard and improved himself and did get so many A-grades, but never 3 in row. The father decided to be more lenient and remove the ‘consecutive’ condition so that the son will get a gift for every three A-grades regardless how spread they are. That means if the son has reached, say **S3** [2 A] and start getting bad grade (**X=0**) again, he will remain remembered that he has still received 2 A’s. On the graph, this is indicated by staying in the same state when **X=0** for **S1**, **S2**, and **S3**. The father built the circuit of Figure 2 and start using it.

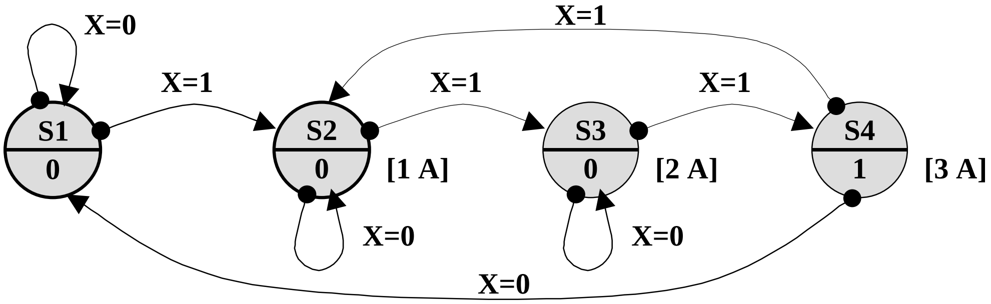


Figure 2. Output Z=1 whenever X=1 for the third time regardless of being spread

3) Soon the father realized that he is now buying too many gifts and at the same time his son is not working any harder than before. He redesigned the circuit so that every bad grade would erase an A-grade. At the same time, to encourage his son to be a straight-A student, the father also enhanced the circuit so that after the third A-grade, the son will get a gift for every single A-grade as long as there is no interruption. Once interrupted by a bad grade, he has to start all over again. The new design is as follows:

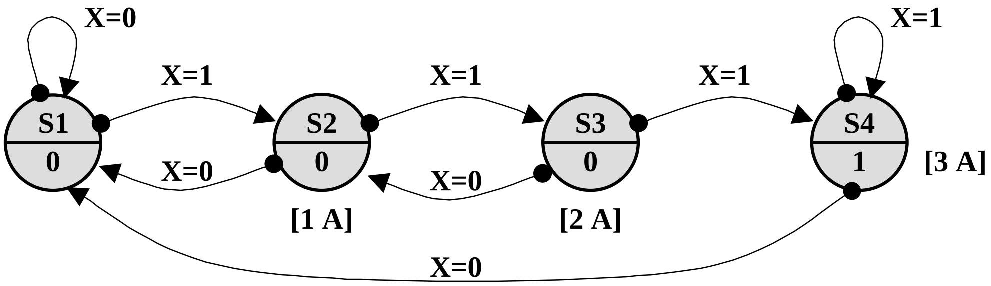


Figure 3. Output Z=1 for X=1 three consecutive times with repeated reward and track-back penalty

♦ You can notice here, that the father has all flexibility to enhance and/or add more restriction. It is only a matter of determining where to go from one state to another for every possible input.

Next State Table, Assignment Table and Next State Maps for Figure 3

Now let us implement the circuit of Figure 3 by first creating the next state table, second assigning binary values to each state, and thirdly draw the next state maps for the required flip flops.

Present State	Next State		Output Z
	X= 0	X= 1	
S1	S1	S2	0
S2	S1	S3	0
S3	S2	S4	0
S4	S1	S4	1

Present State	Next State		Output Z
	X= 0	X= 1	
00	00	01	0
01	00	11	0
11	01	10	0
10	00	10	1

Q1Q2 X	0	1
00	0	0
01	0	1
11	0	1
10	0	1

Q1+

	0	1
00	0	1
01	0	1
11	1	0
10	0	0

Q2+

Input Excitation Maps

The next step in the design after the next state maps are determined is to determine the input excitation tables, which are specific to certain flip-flops. Let us start the design with a DFF and then TFF and JKFF.

Q1Q2 X	0	1
00		
01		
11		
10		

D1

D1= ____________

Q1Q2 X	0	1
00		
01		
11		
10		

D2

D2= ____________

Q1Q2 X	0	1
00		
01		
11		
10		

T1

T1= ____________

Q1Q2 X	0	1
00		
01		
11		
10		

T2

T2= ____________

Q1Q2 X	0	1
00		
01		
11		
10		

J1

	0	1
00		
01		
11		
10		

K1

J1= ____________

K1= ____________

Q1Q2 X	0	1
00		
01		
11		
10		

J2

	0	1
00		
01		
11		
10		

K2

J2= ____________

K2= ____________

State Graph Design (Mealy Style)

In the Moore style (Figures 1, 2 and 3), the output **Z** is mad as a function of the present state **S** only and therefore, the output **Z** is attached to it and placed within the state circle. Whereas in the Mealy style, the output **Z** is a function of both the present state **S** and the input **X**. Therefore, the output **Z** is attached to the transition arrows and placed along with input **X** on the arrows as a pair **X/Z**. The following are the above three designs (Figures 1, 2 and 3) redrawn in Mealy style.

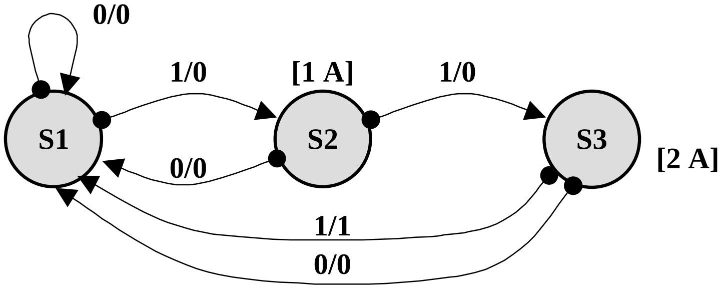


Figure 4. Output Z=1 if and only if X=1 three consecutive times, else start all over again

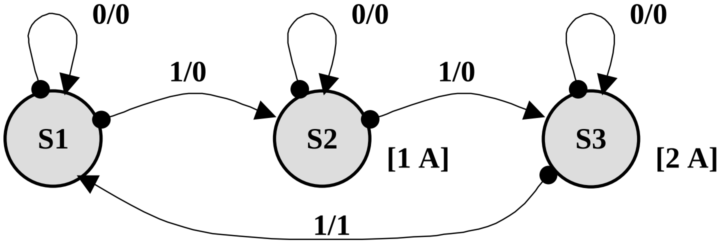


Figure 5. Output Z=1 whenever X=1 for the third time regardless of being spread

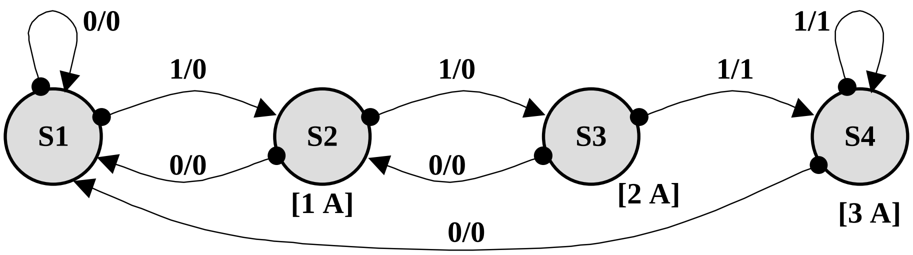


Figure 6. Output Z=1 for X=1 three consecutive times with repeated reward and track-back penalty

Next State Table, Assignment Table and Next State Maps for Figure 5

Now let us implement the circuit of Figure 5 (Mealy) by first creating the next state table, second assigning binary values to each state, and thirdly draw the next state maps for the required flip flops. Note that this sequential machine has only 3 states. So when we assign binary values for 2 flip-flops, the unused states are filled with don’t cares.

Present State	Next State		Output Z	
	X= 0	X= 1	X= 0	X= 1
S1	S1	S2	0	0
S2	S2	S3	0	0
S3	S3	S1	0	1

Present State	Next State		Output Z	
	X= 0	X= 1	X= 0	X= 1
00	00	01	0	0
01	01	11	0	0
11	11	00	0	1
10	xx	xx	x	x

Q1+

Q1Q2 \ X	0	1
00	0	0
01	0	1
11	1	0
10	x	x

Q2+

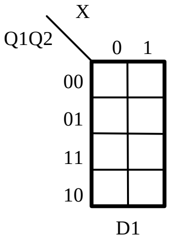
Q1Q2 \ X	0	1
00	0	1
01	1	1
11	1	0
10	x	x

Z

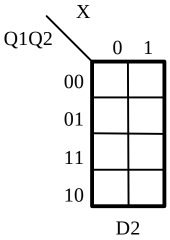
Q1Q2 \ X	0	1
00	0	0
01	0	0
11	0	1
10	x	x

Input Excitation Maps

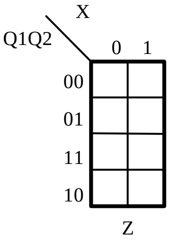
The next step in the design after the next state maps are determined is to determine the input excitation tables, which are specific to certain flip-flops. Let us start the design with a DFF and then TFF and JKFF. Note that output Z is the same for all three different implementations



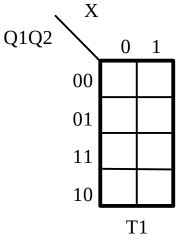
D1= _____________



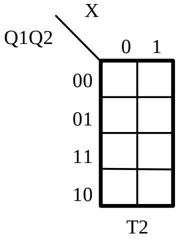
D2= _____________



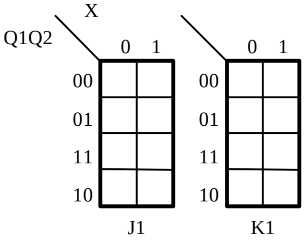
Z= _____________



T1= _____________

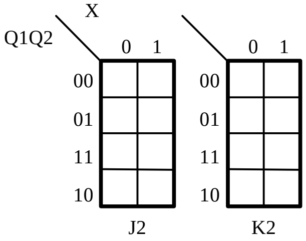


T2= _____________



J1= _____________

K1= _____________



J2= _____________

K2= _____________