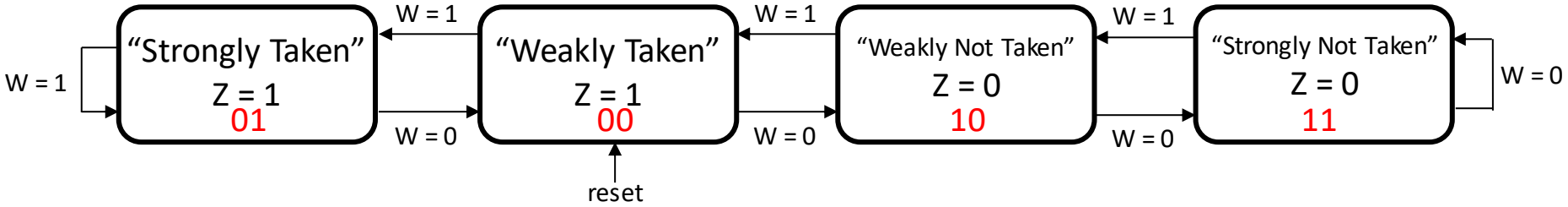
Module 75:

- A common instruction for a CPU is a “branch”, which will change the flow of a program’s execution based on some condition. Branches can either be “taken” or “not taken”. In order to increase performance, modern CPUs often attempt to predict the result of a branch ahead of time.
- Below is a state diagram of a common 2-bit branch predictor. The output Z represents our next prediction: 1 if we predict taken, 0 if we predict not taken. The input W represents if the branch was actually taken or not taken the last time it was executed.
- Create a state table from this state diagram. Use the state assignments shown in red. Use K-maps to find the minimum cost logic expressions. Then, draw the a circuit from your table. Use D flip-flops in your design.



Module 76:

- Given this state table, try the two different assignments and see which results in lower cost logic expressions for Y_1, Y_0, and z.
- Create state assignment tables and then the logic expressions for each.
- Use K-maps to find the minimum cost logic expressions.
- Assignment 1: A = 00, B = 01, C = 10, D = 11
- Assignment 2: A = 11, B = 10, C = 01, D = 00

Present State	Next State		Output z
	w = 0	w = 1	
A	B	C	0
B	A	C	0
C	B	D	0
D	A	B	1

Module 77: Convert the this state table into an equivalent FSM with hot-encoded states. Then derive logical functions for the “next state” and output variables (you don’t need to draw the circuit).

Present state y_1y_0	Next State Y_1Y_0		Z
	W = 0	W = 1	
00	00	01	0
01	10	01	0
10	00	11	0
11	10	01	1