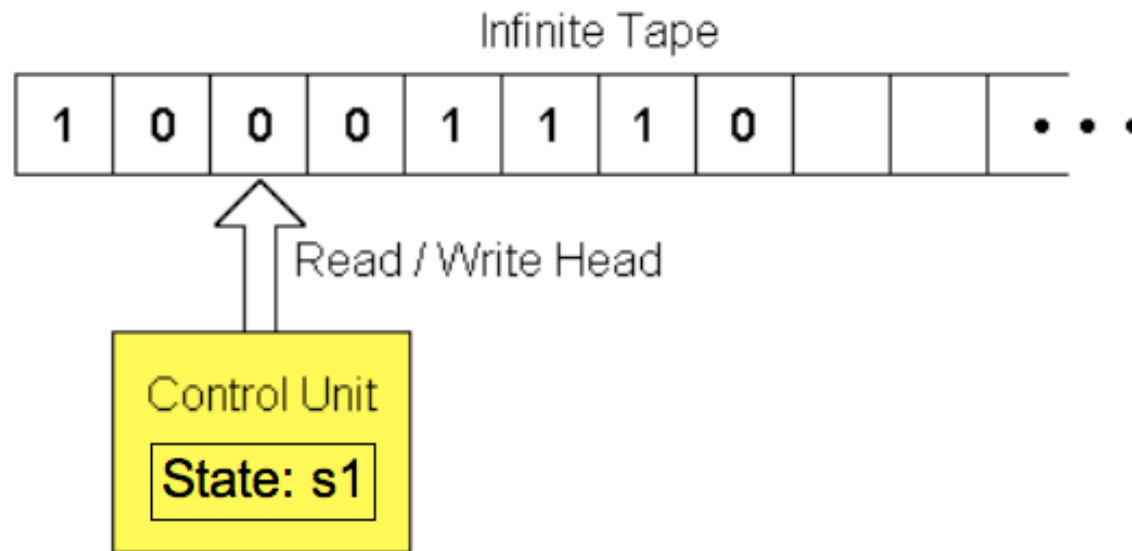


Turing Machines

Turing Machines



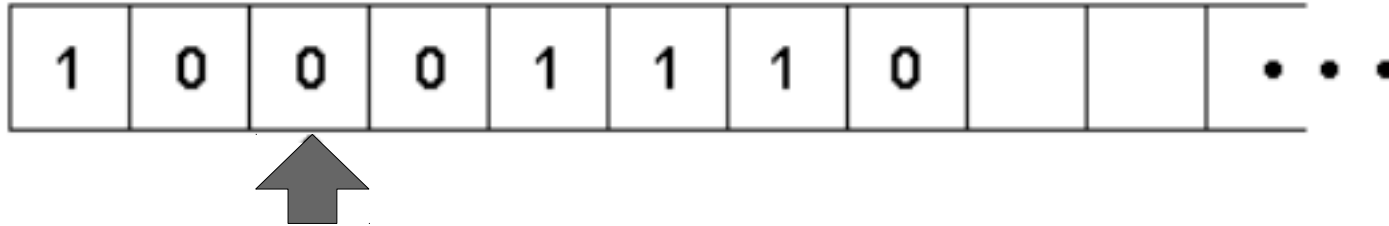
- A Turing machine consists of:
 - A finite set of **states** that it can be in
 - A finite set of **symbols** that it can read and write
 - An infinite **tape** divided into cells for storing symbols
 - A finite **rule table** that tells the machine what to do

States

- How does a **vending machine** “know” whether you have put in the correct amount of change?
- Sometimes when you press a button, nothing happens
- Other times when you press the same button, the machine gives you a soda
- The machine can be in one of several possible internal **states**:
 - “Insufficient money received”
 - “Exact amount received”
 - “Too much money received”

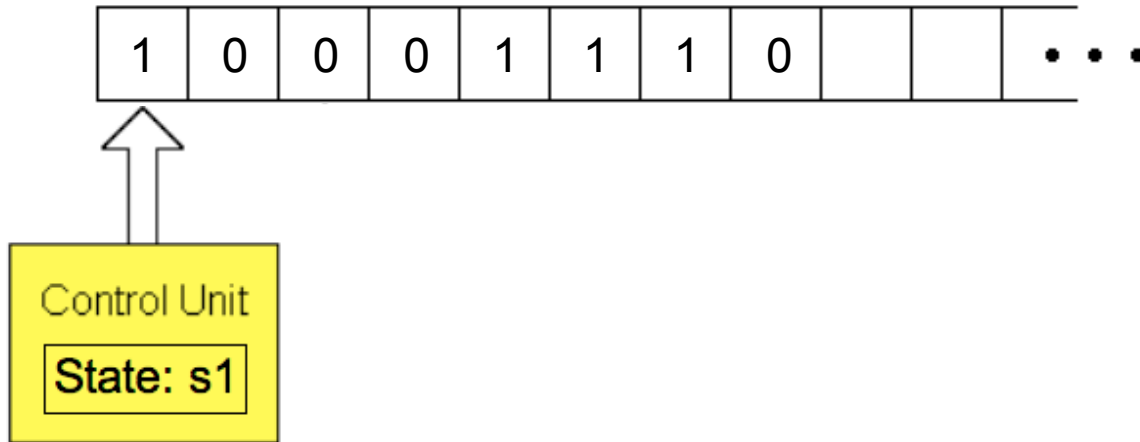


Symbols and the Tape



- Symbols are usually just **0**, **1**, and **blank**
- Any arbitrary set of symbols can be used
... as long as the set is **finite**
- The tape stores **one symbol per cell**
... but has an **infinite number of cells** available
- The tape head **reads or writes** one symbol at a time
... and then moves **left, right**, or makes **no move**

An Example

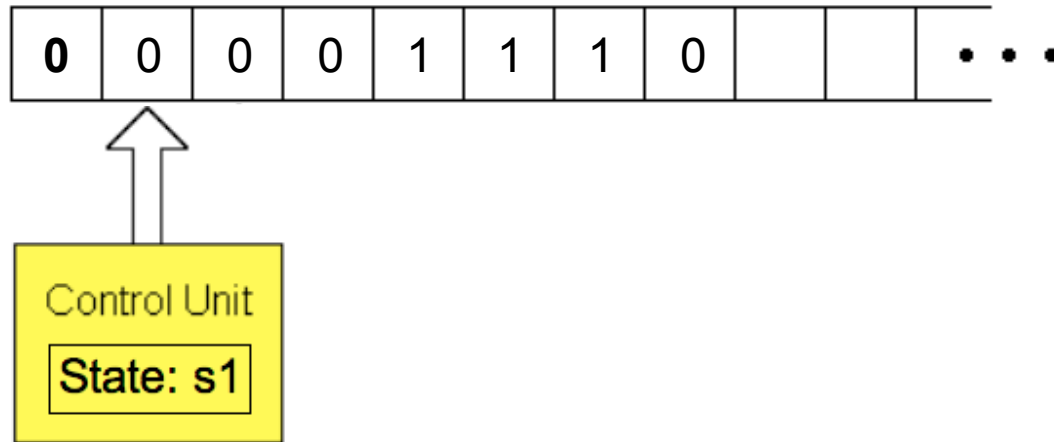


States: **s1, halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

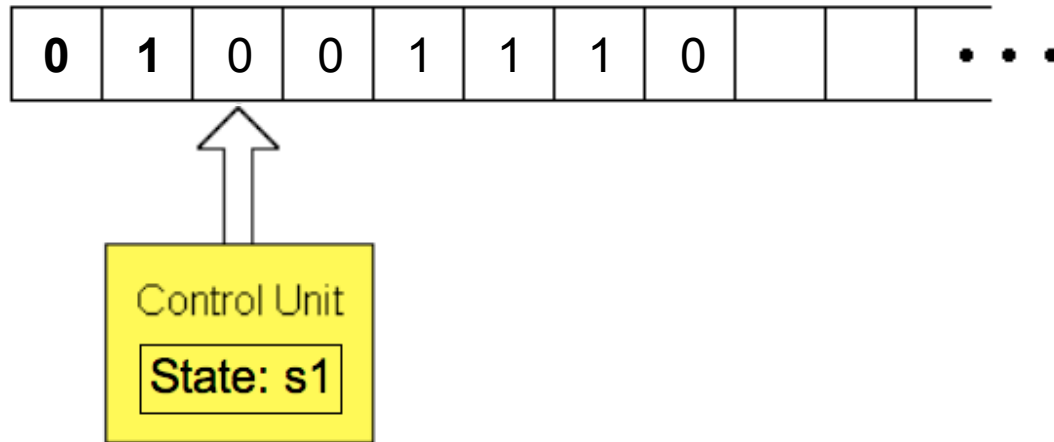


States: **s1, halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

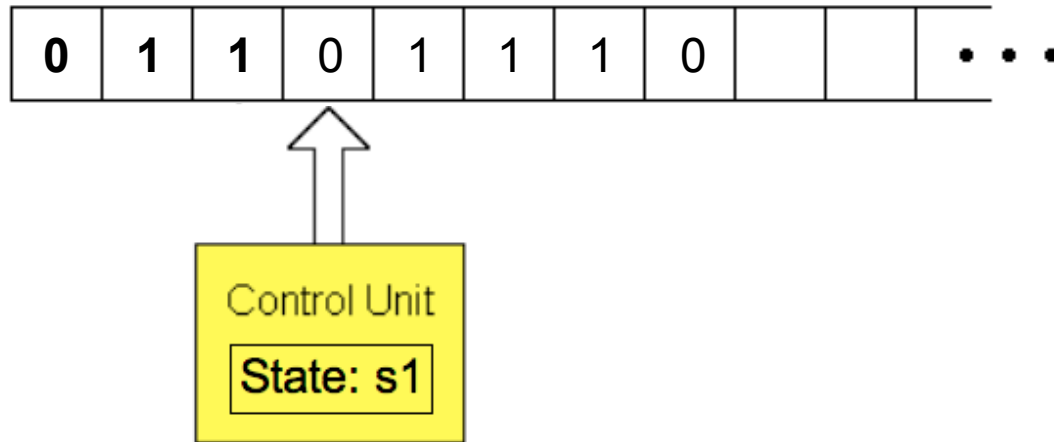


States: **s1**, **halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

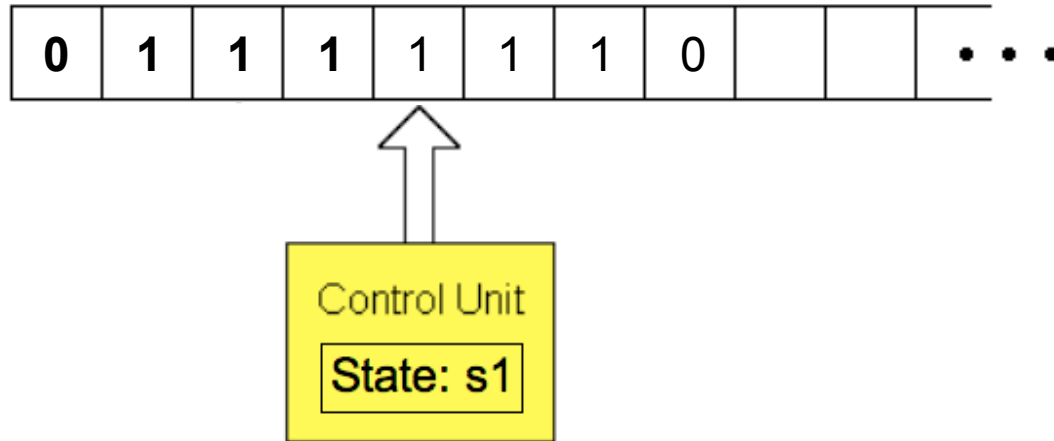


States: **s1**, **halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

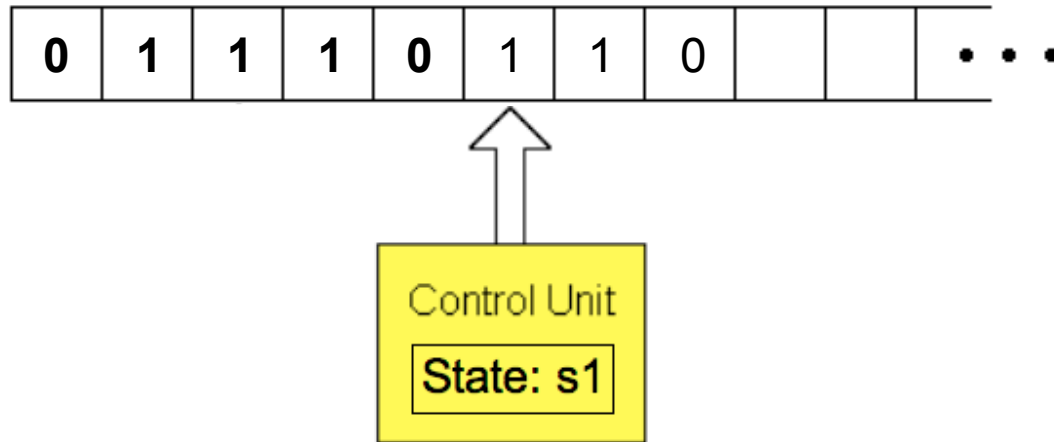


States: **s1, halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

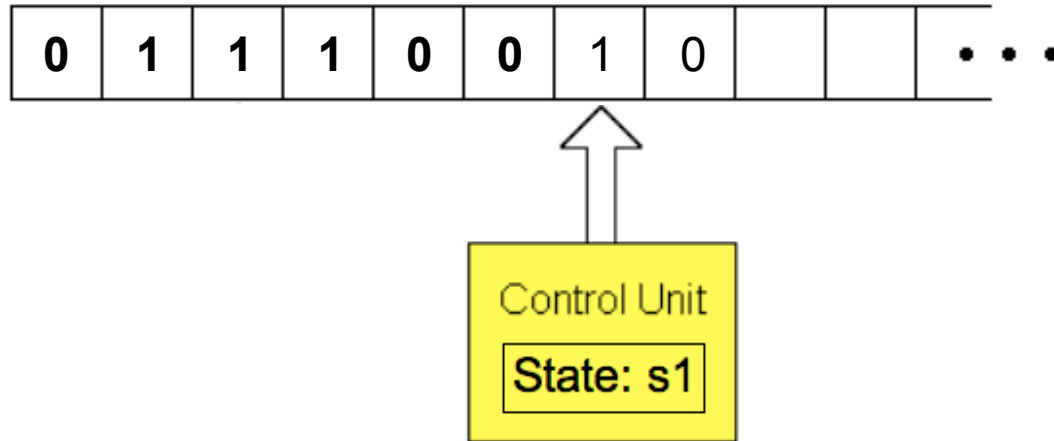


States: **s1**, **halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

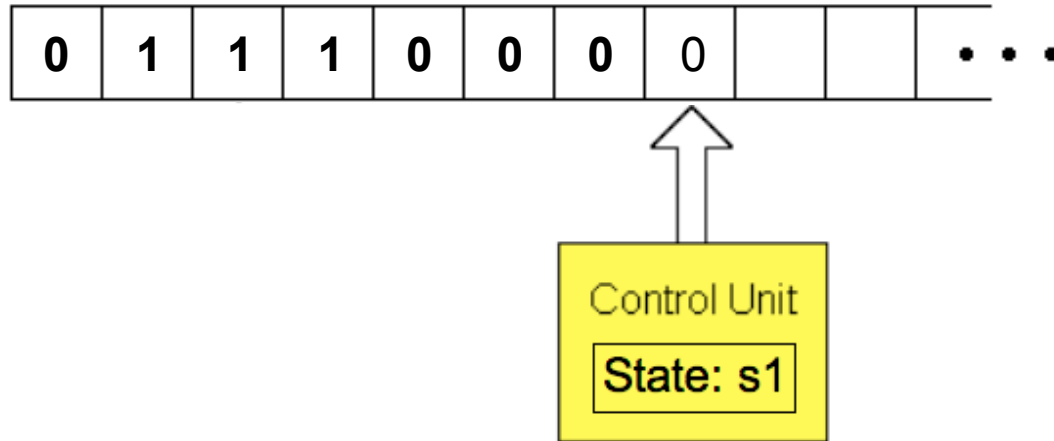


States: **s1**, **halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

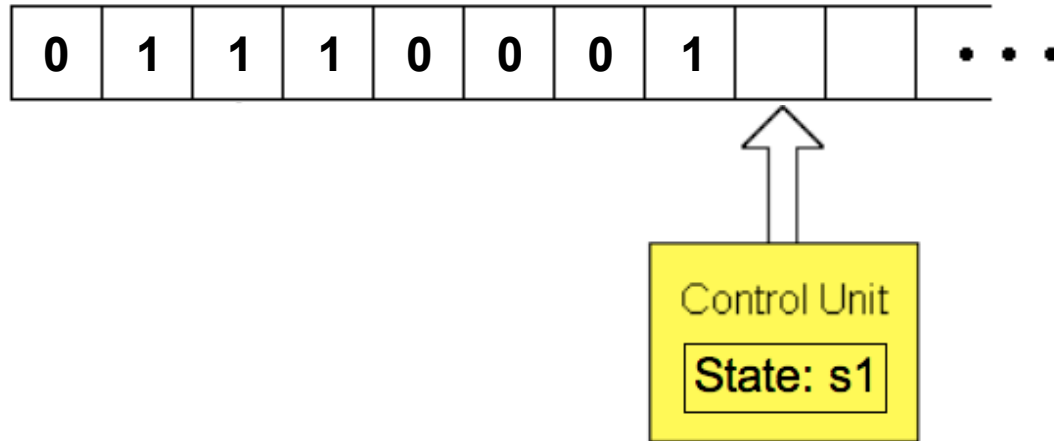


States: **s1, halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

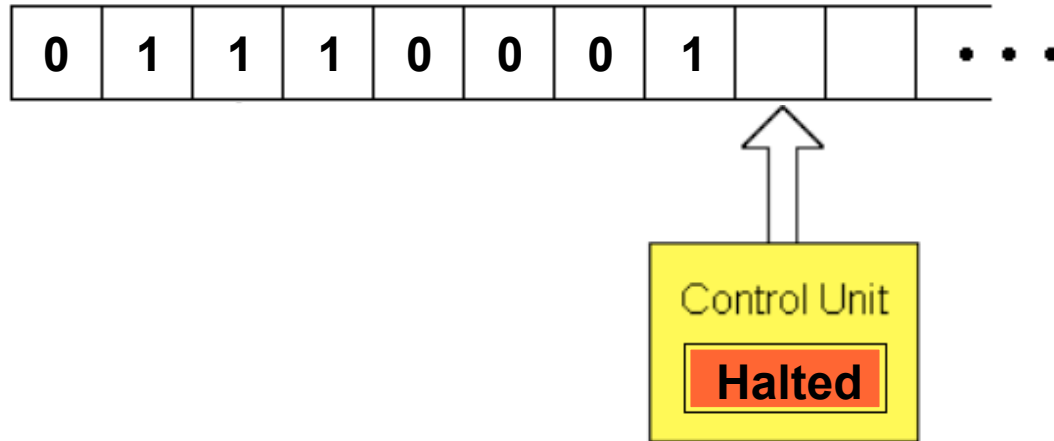


States: **s1**, **halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example

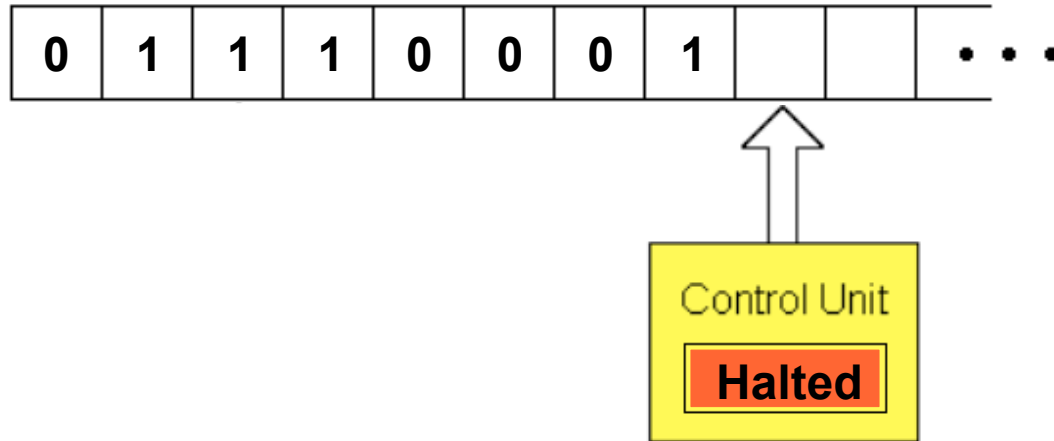


States: **s1, halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	1	right	s1
s1	1	0	right	s1
s1	blank	blank	none	halt

An Example



This machine performs the “binary inversion” function

10001110 → 01110001

000 → 111

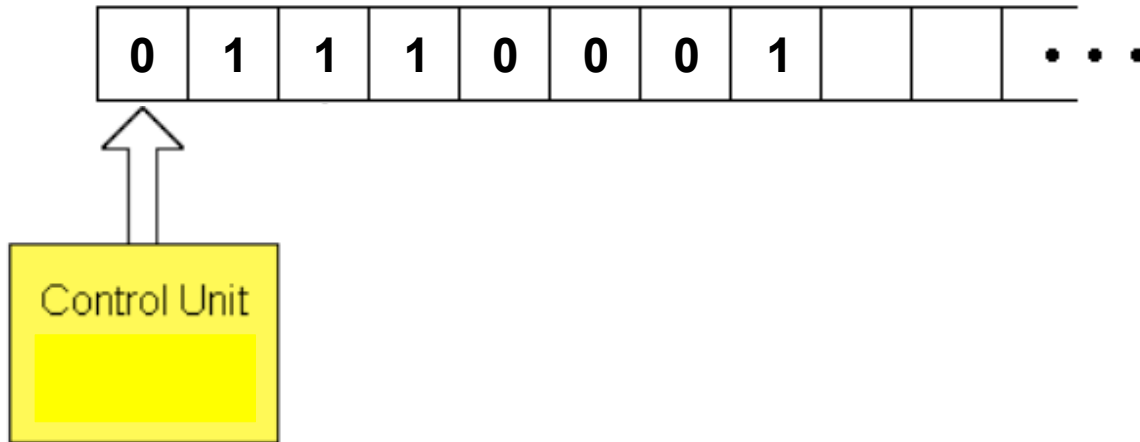
1100 → 0011

1 → 0

0101010 → 1010101

etc...

Your Turn: Design an “Eraser” Machine

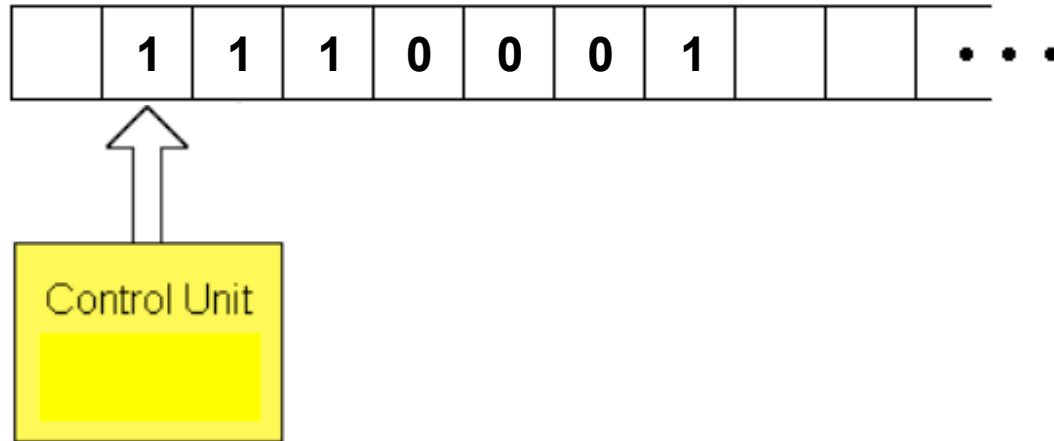


States:

Rules:

current state current symbol new symbol move new state

Your Turn: Design an “Eraser” Machine

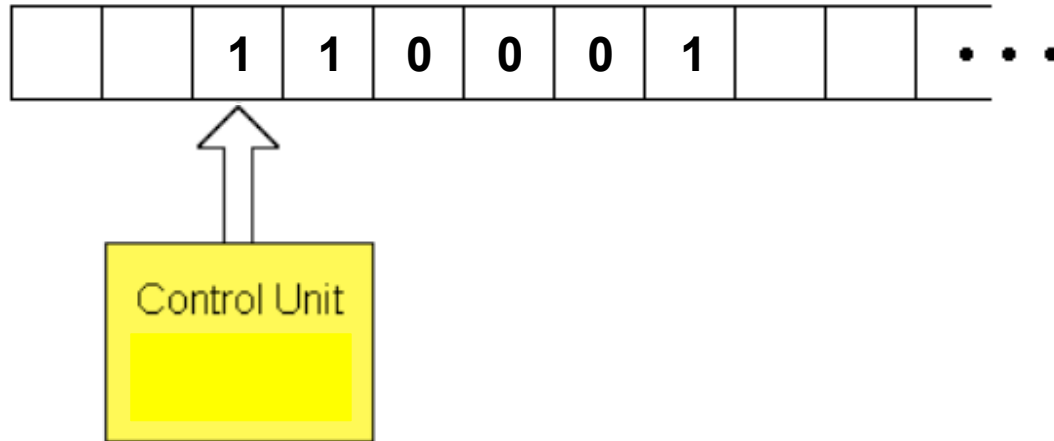


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

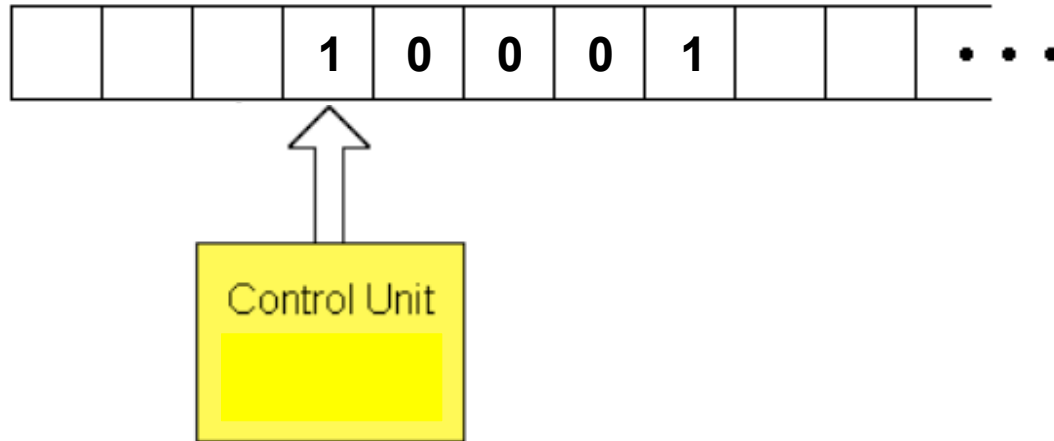


States:

Rules:

current state current symbol new symbol move new state

Your Turn: Design an “Eraser” Machine

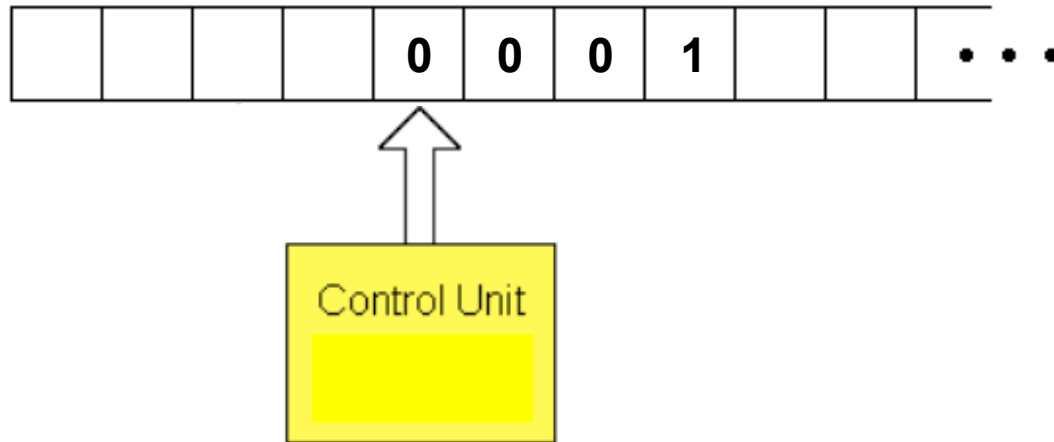


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

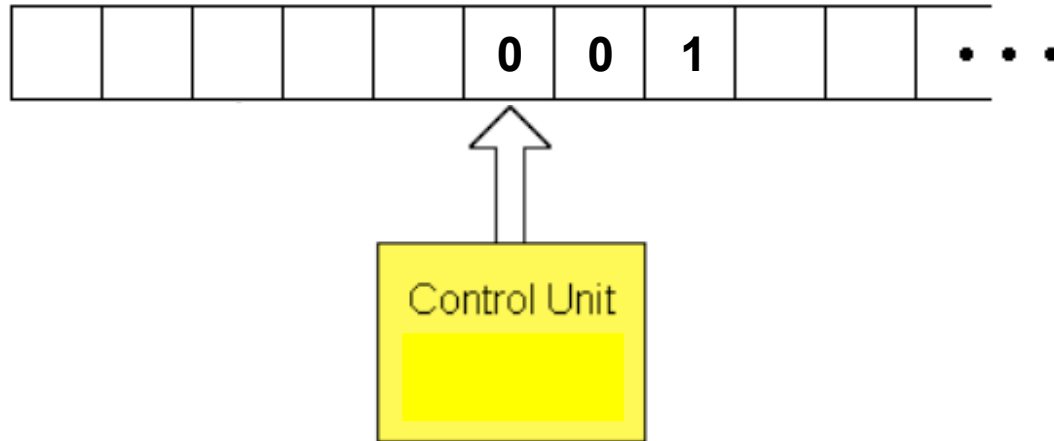


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

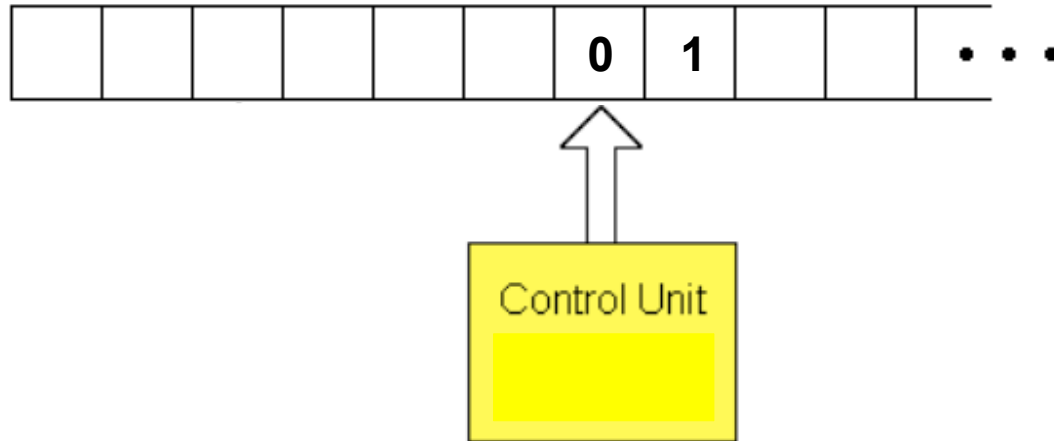


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

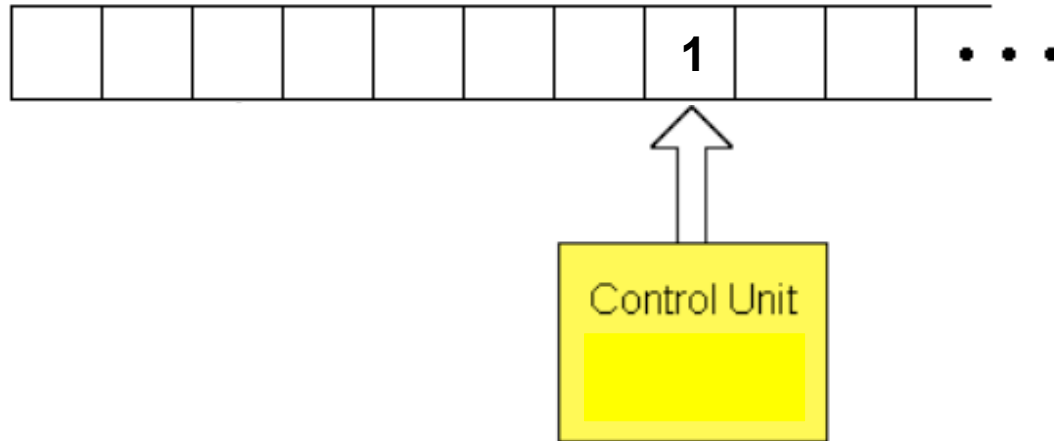


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

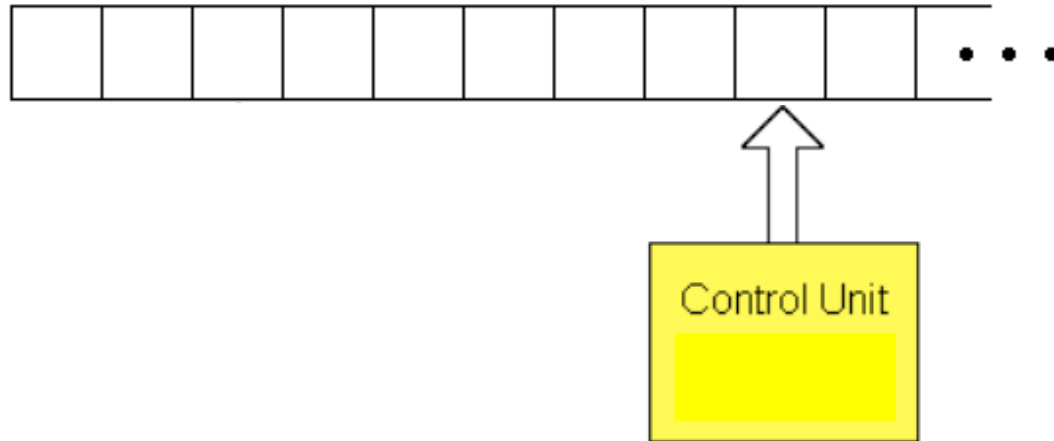


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

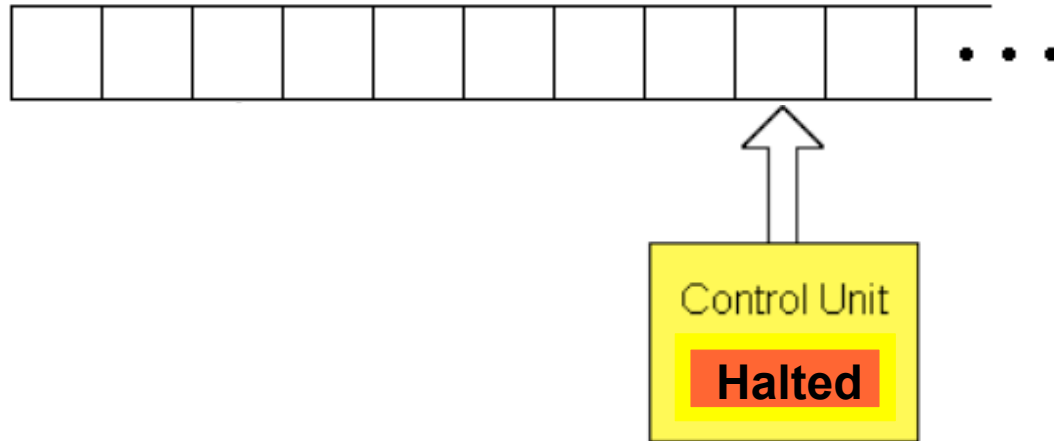


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

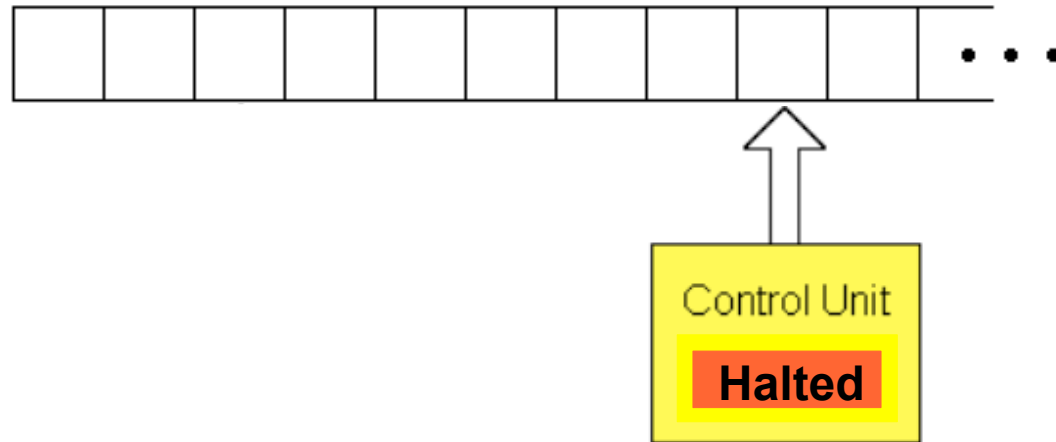


States:

Rules:

current state *current symbol* *new symbol* *move* *new state*

Your Turn: Design an “Eraser” Machine

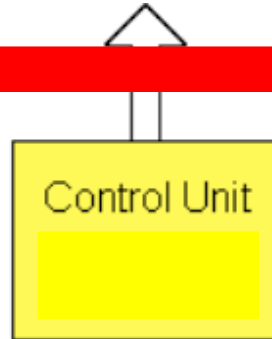
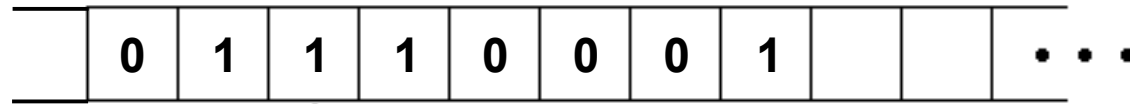


States: **s1, halt**

Rules:

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	<i>blank</i>	right	s1
s1	1	<i>blank</i>	right	s1
s1	<i>blank</i>	<i>blank</i>	none	halt

How About a “Zigzag” Machine?

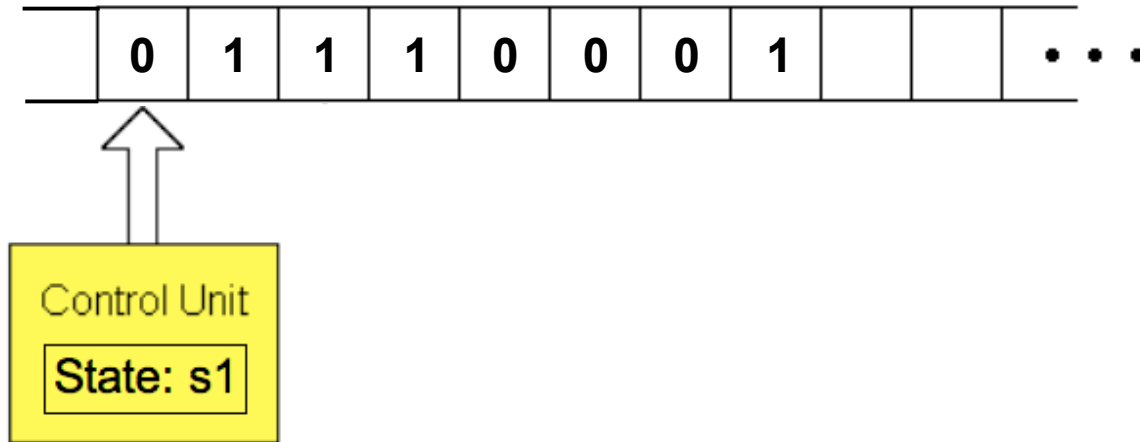


Tape head moves endlessly back and forth from one end of the string to the other, without changing any symbols

States: **s1, s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

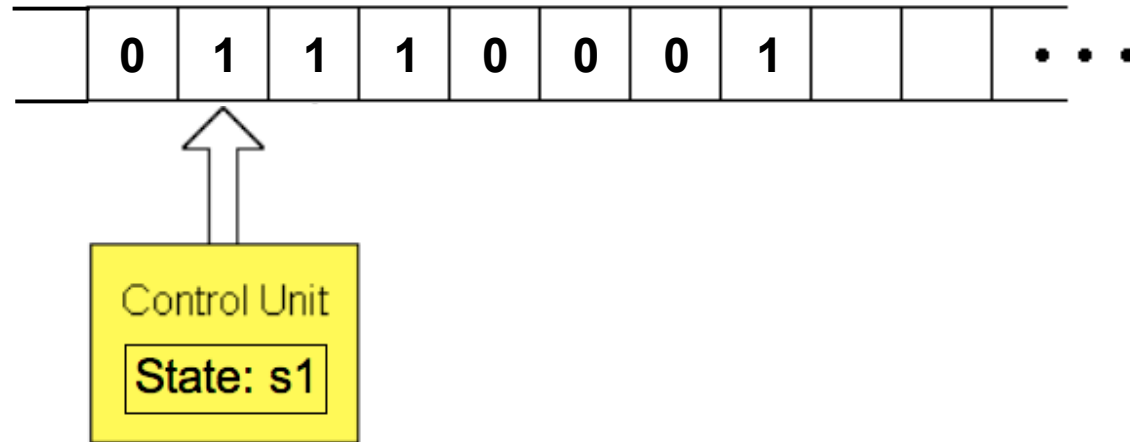
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

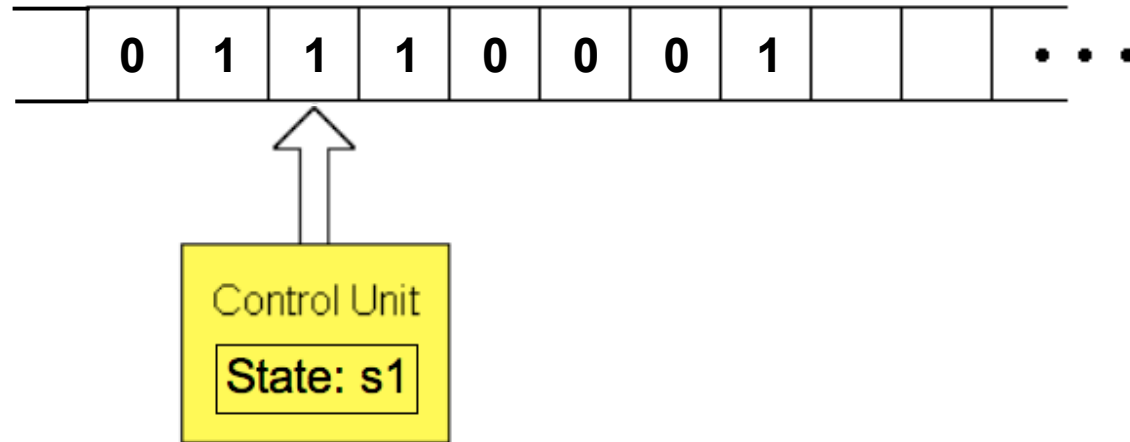
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

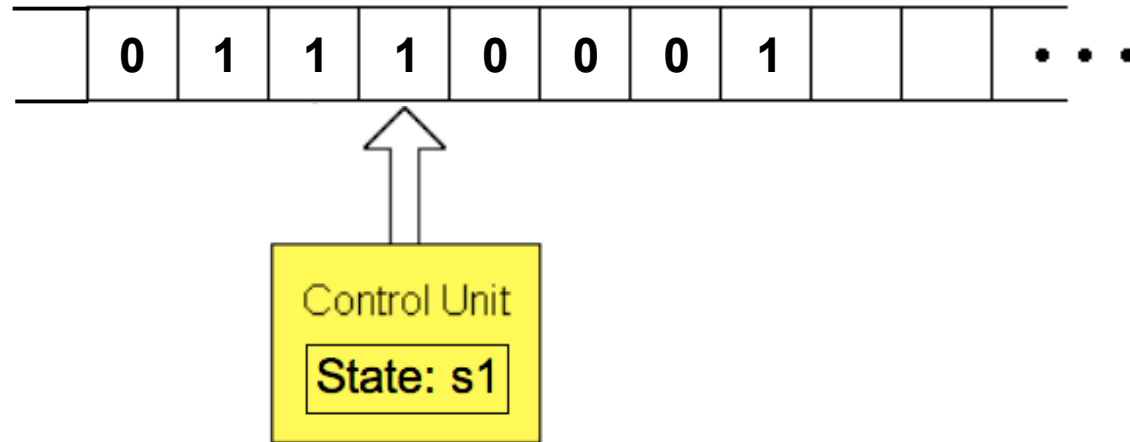
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

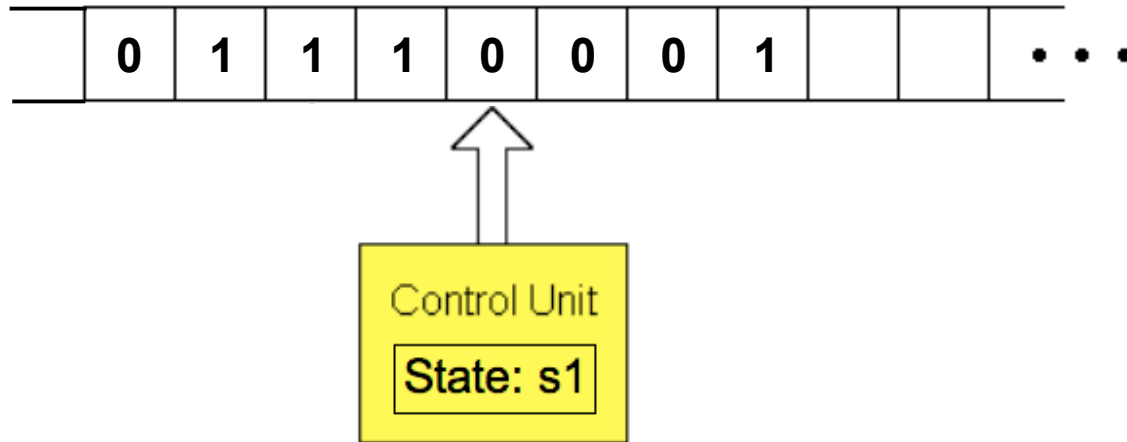
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

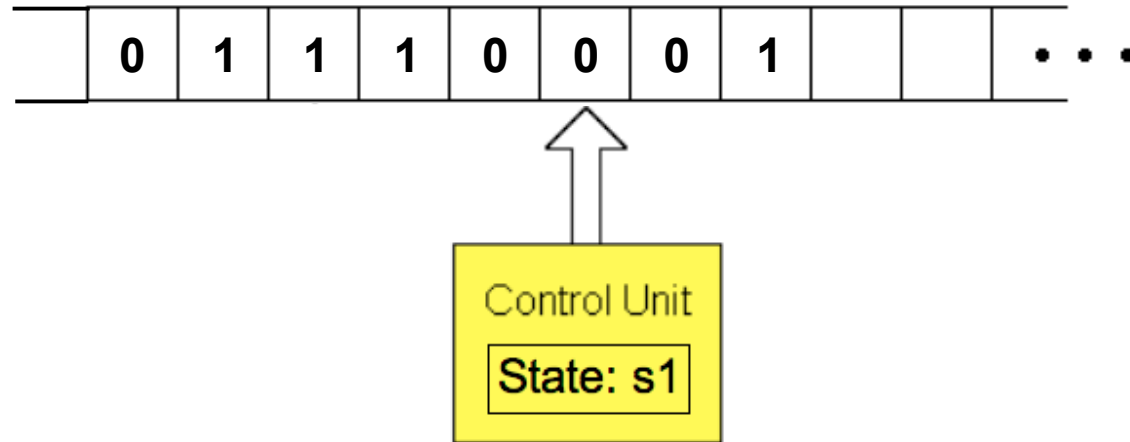
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

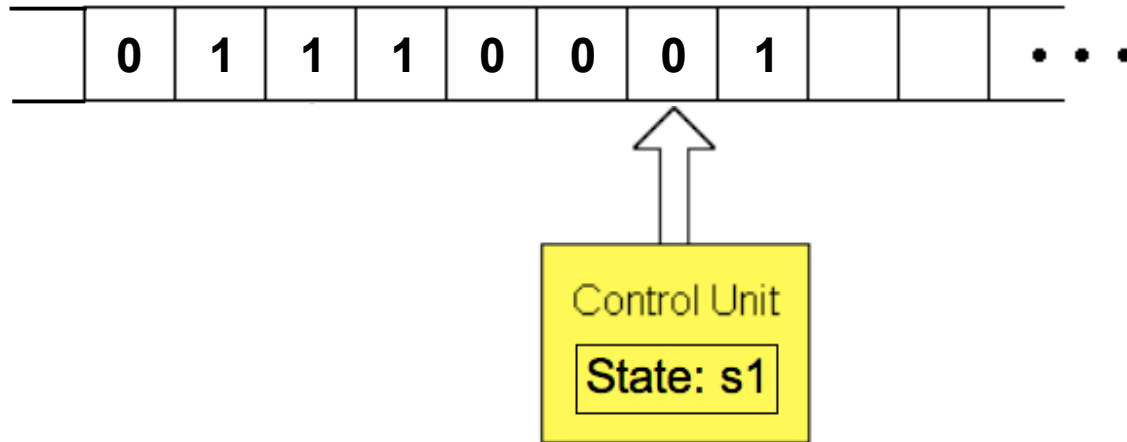
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

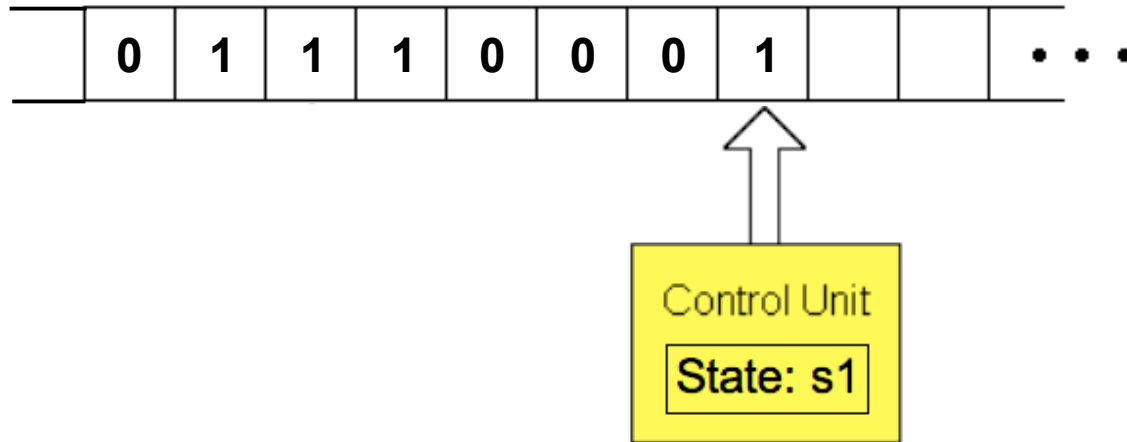
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

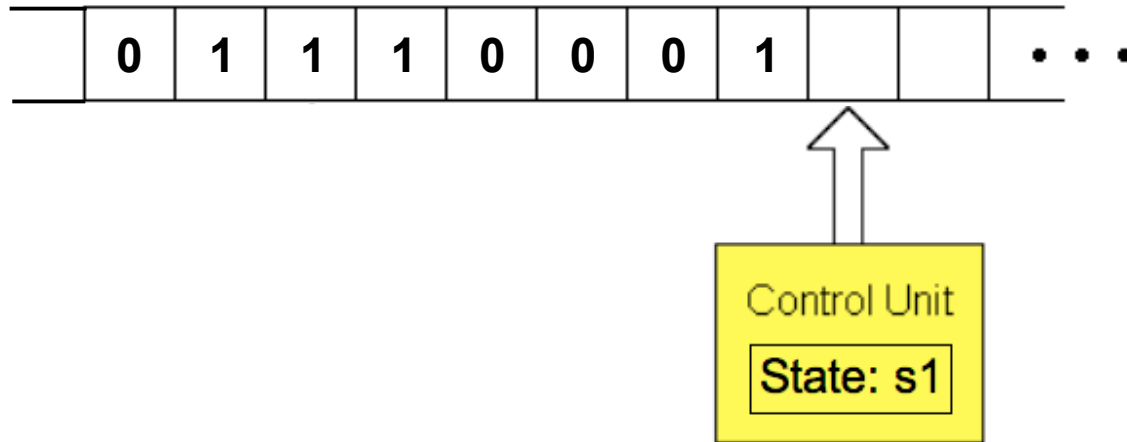
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

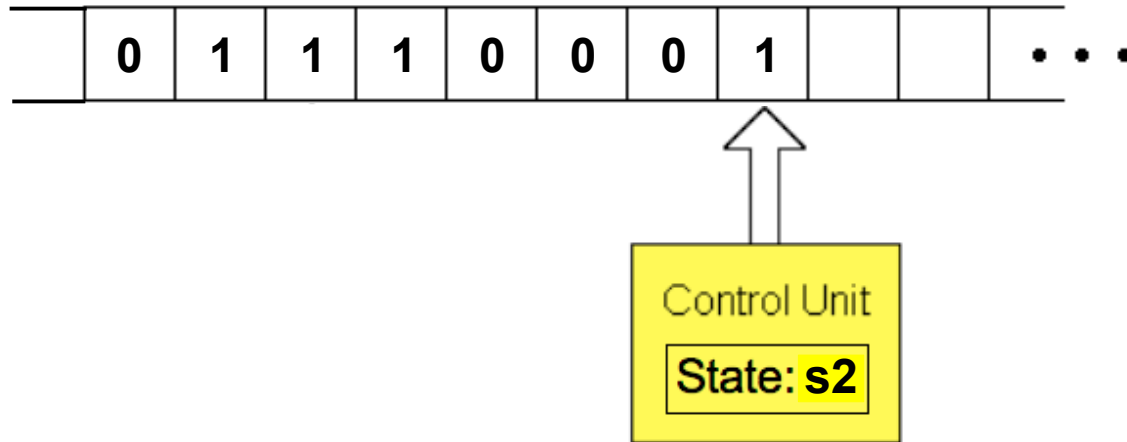
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

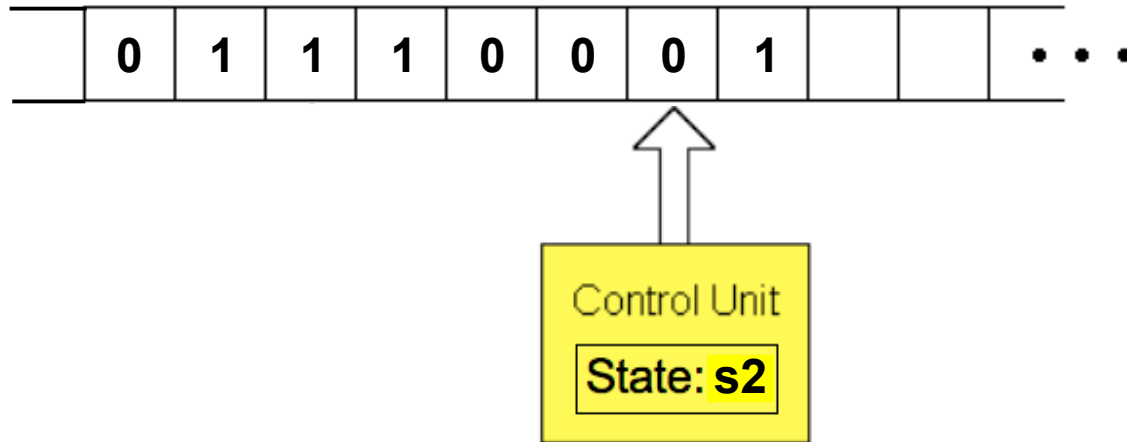
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

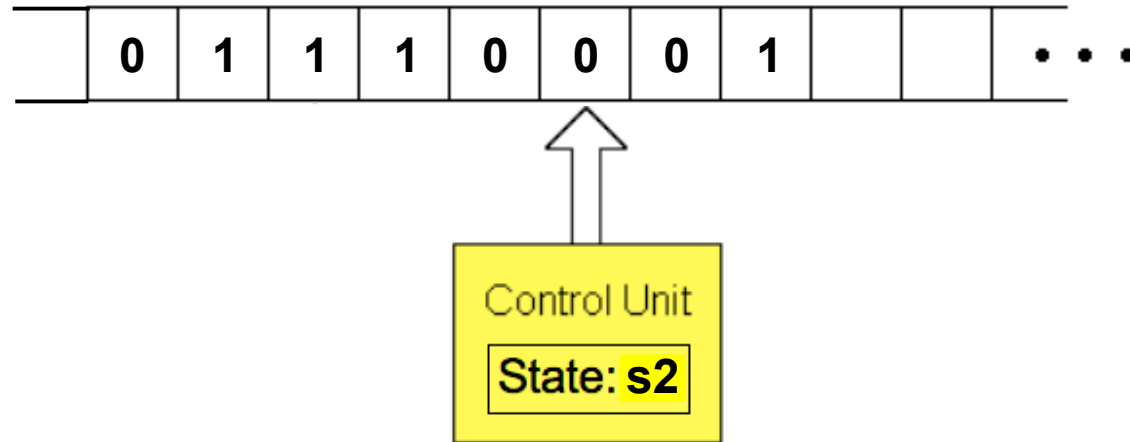
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

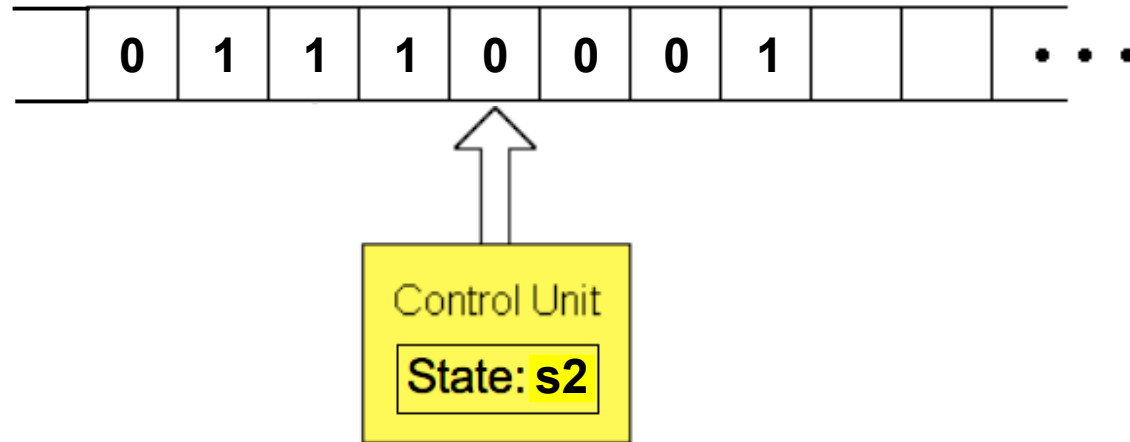
How About a “Zigzag” Machine?



States: **s1, s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

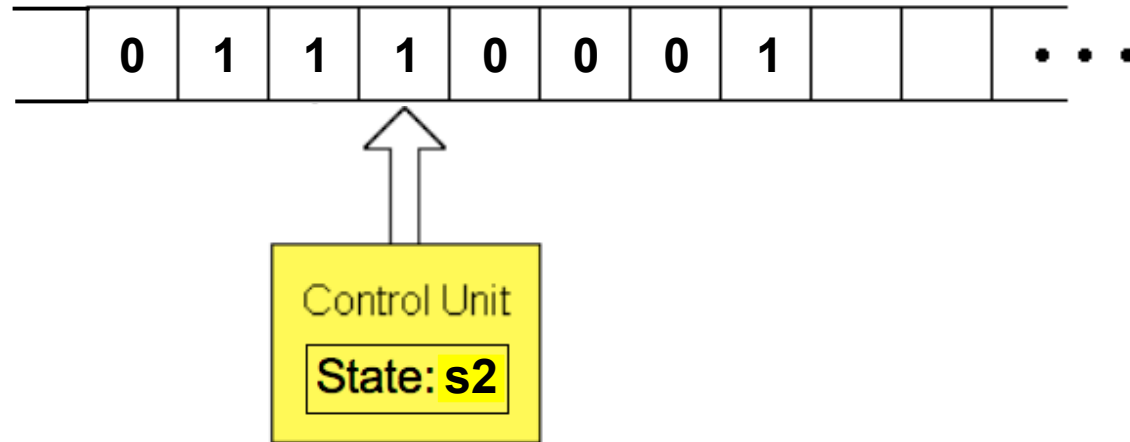
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

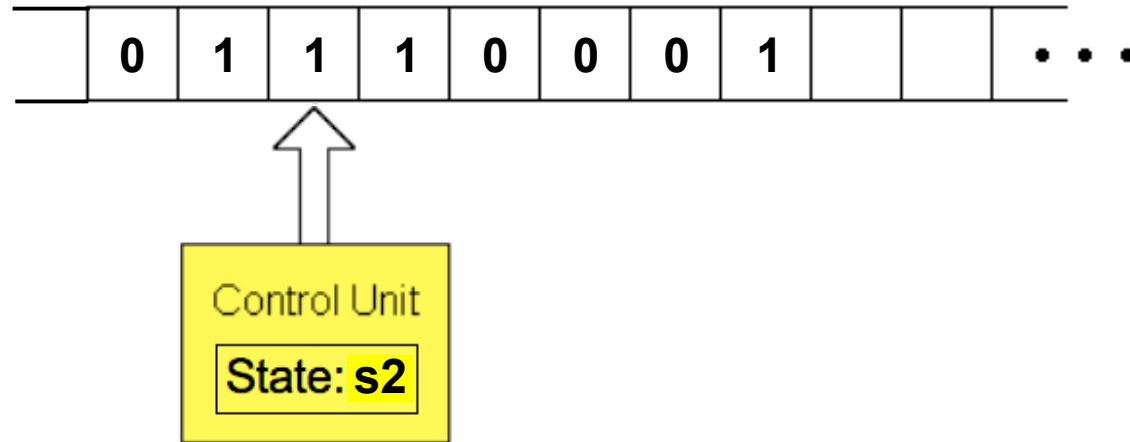
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

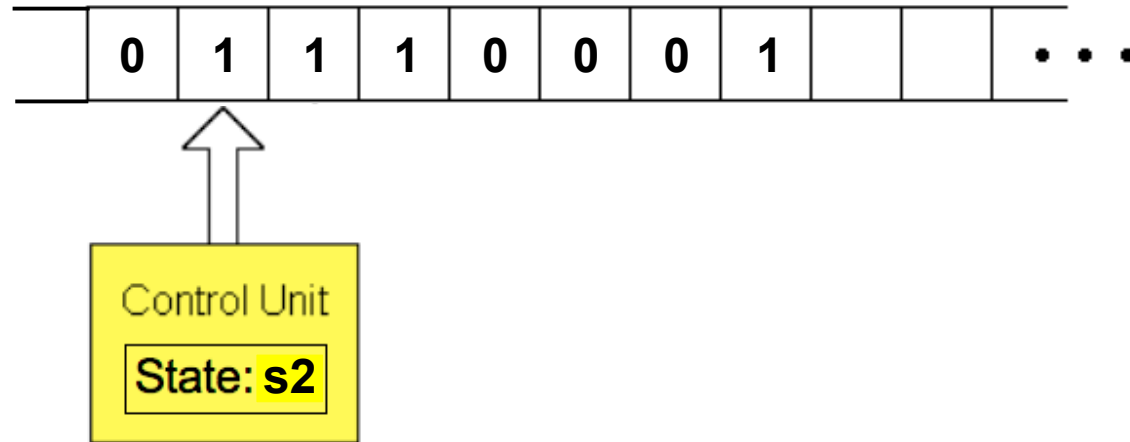
How About a “Zigzag” Machine?



States: **s1, s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

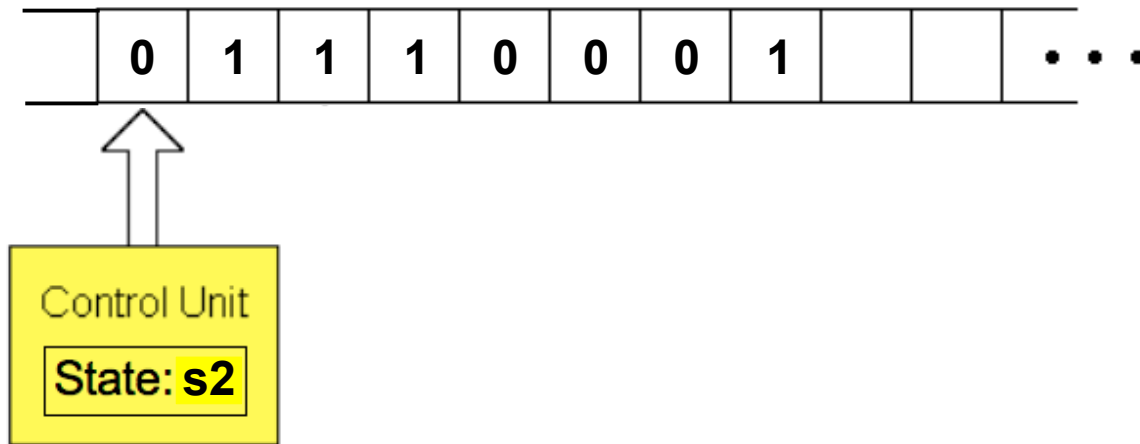
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

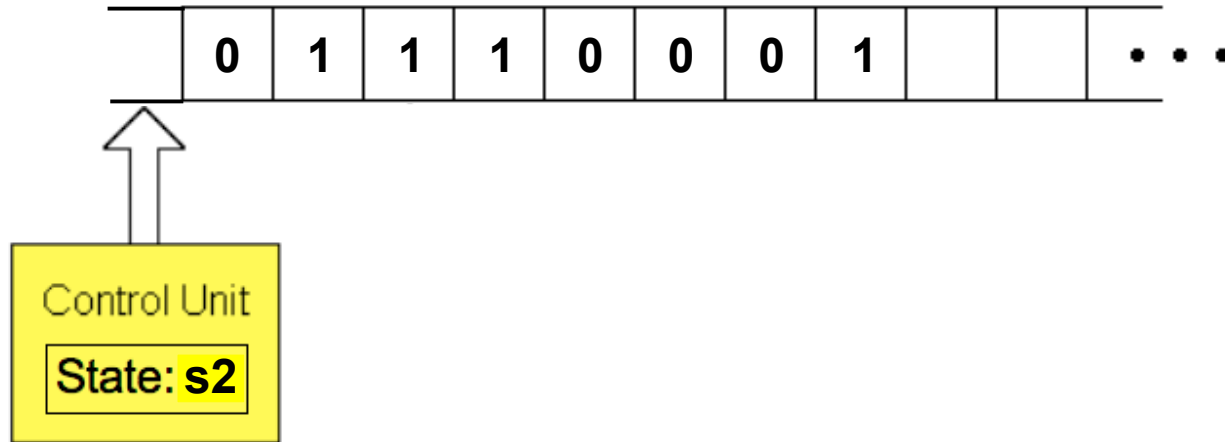
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

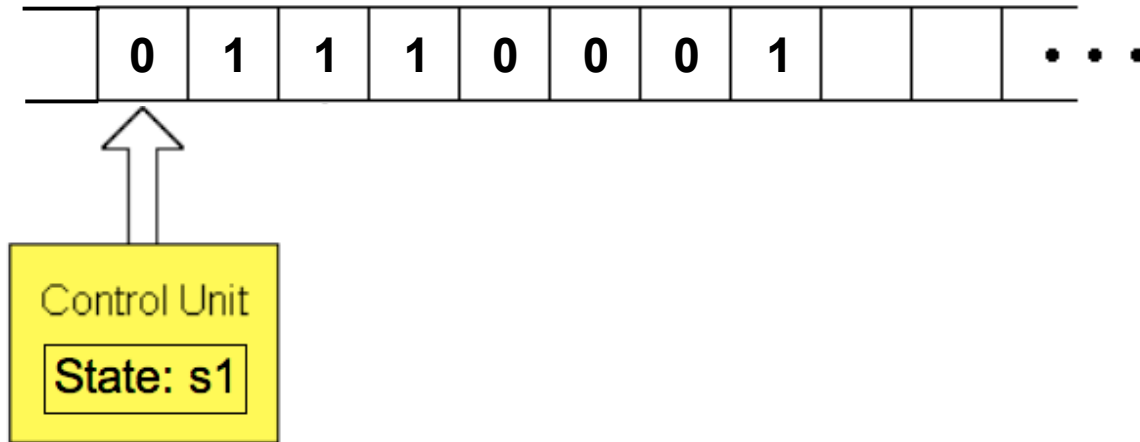
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

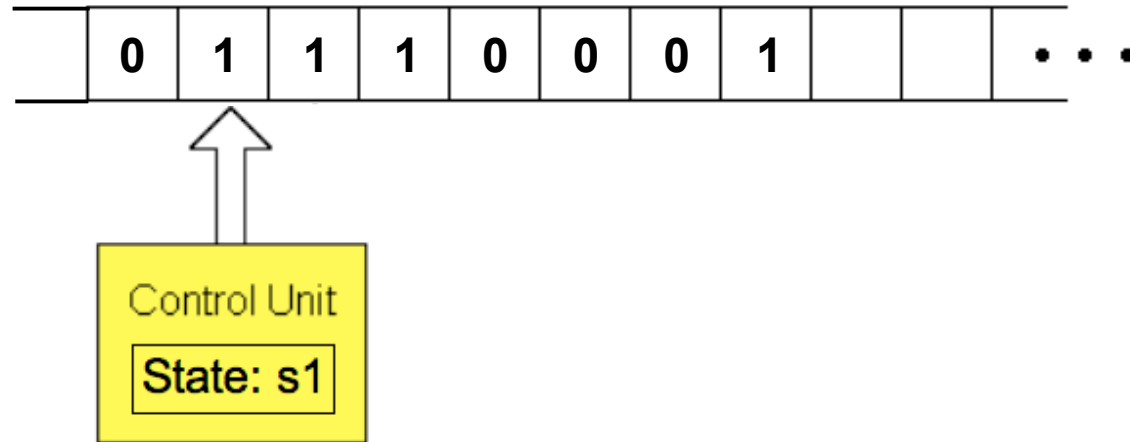
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

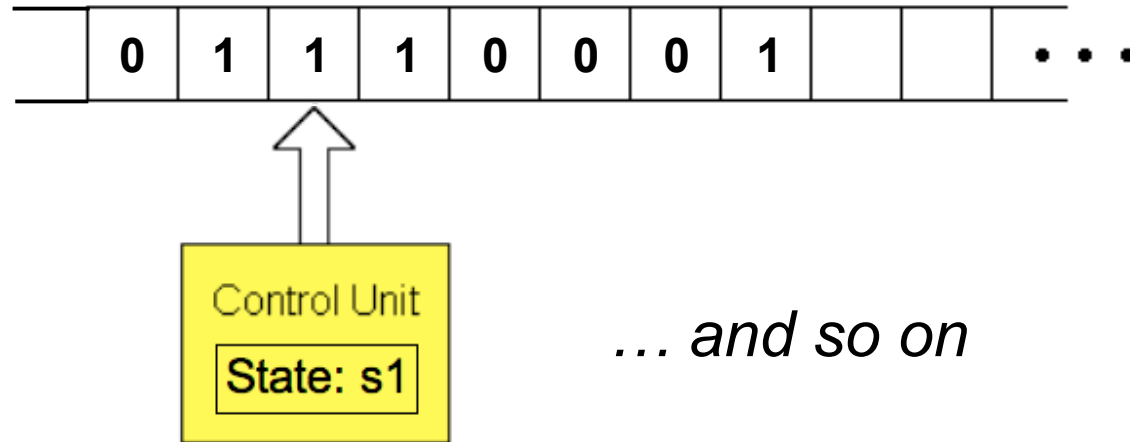
How About a “Zigzag” Machine?



States: **s1**, **s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1

How About a “Zigzag” Machine?



States: **s1, s2**

<i>current state</i>	<i>current symbol</i>	<i>new symbol</i>	<i>move</i>	<i>new state</i>
s1	0	0	right	s1
s1	1	1	right	s1
s1	blank	blank	left	s2
s2	0	0	left	s2
s2	1	1	left	s2
s2	blank	blank	right	s1