EECS 16A Designing Information Devices and Systems I
Fall 2015 Anant Sahai, Ali Niknejad

Discussion 7B

1. Analog to Digital Conversion

In many applications, it is useful to convert an "analog" voltage into a "digital" representation of that voltage. There is a circuit that performs this operation aptly named the Analog-to-Digital converter (ADC). One type of ADC is the Successive Approximation Register ADC, which uses a method called "binary search" to efficiently find the digital representation of an input voltage.

An ADC allows us to take a real, physical signal, like the solar cell voltage in the imaging lab or the mesh voltage in the resistive touch screen lab, and turn it into bits that can be interpreted and manipulated by a computer. Another example is an infrared sensor that measures the distance of an object to the sensor and outputs a voltage in some linear range. This voltage is input into the MSP430, converted to a digital representation, sent to a computer, and then calculated by the computer back to the distance of the object.

(a) **Quantization**

Quantization is the process of estimating a continuous value (like an analog signal) by setting limits on the precision you have for that value. If you've taken chemistry, you can think of it as "limiting the number of significant figures." When converting to digital, it is necessary to convert to a binary representation. In general, how do you represent an arbitrary number using binary?

(b) **Binary Search**

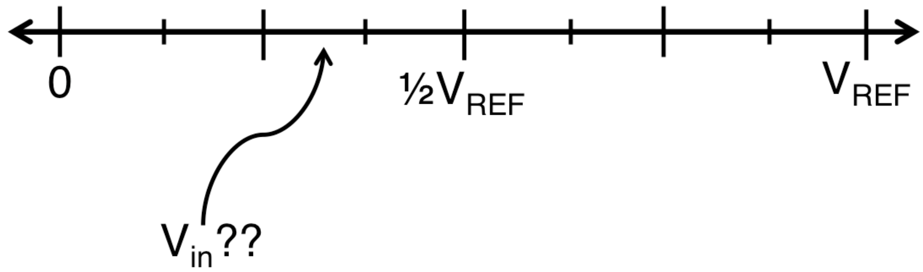
Pseudocode for Binary Search:

```
def binary_search(value, lst):
    first = 0
    last = len(lst) - 1

    while first<=last:
        midpoint = (first+last)//2
        if value == lst[midpoint]:
            return midpoint
        elif value < lst[midpoint]:
            last = midpoint - 1
        else:
            first = midpoint + 1

    return "Value not found in list!"
```

Because we're working with an analog signal, we aren't given a list of discrete bins to find the signal (the way binary search works). Instead, we divide up a range of possible values into bins, and find the analog signal in those bins. The range of possible values is the **dynamic range** of the ADC, and is $[0, V_{ref}]$. This means that if the input voltage is greater than V_{ref}, the ADC won't be able to tell us anything (beyond returning a string of all 1's).

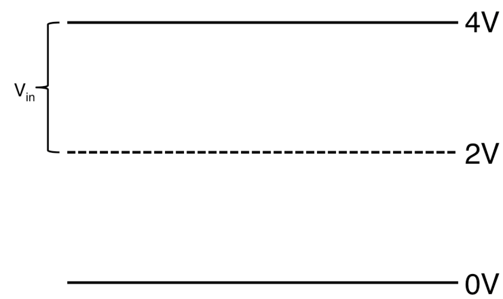


We will do the same thing as the binary search algorithm, except "dividing the range into bins" happens as the algorithm progresses. Every time we choose the left or the right half of the effective number line we'll log a 1 or 0 in our "binary representation". We stop when we have N number of bits (digits) in the binary representation, determined by the **resolution** of the SAR ADC. For example, the MSP 430 you use in lab has a 12-bit SAR ADC, or an ADC with a resolution of 12.

Let's walk through the ADC Binary Search algorithm for the value 2.46, with reference voltage 4V and a 4-bit SAR ADC.

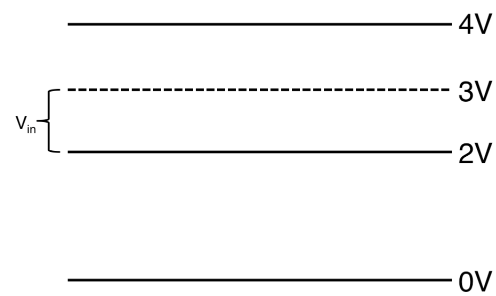
Step 1 (finding the **MSB**):

- Minval = 0
- Maxval = 4
- binary_representation = ""
- Midpoint = 2
- 2.46 > 2 $\rightarrow$ binary_representation += "1"



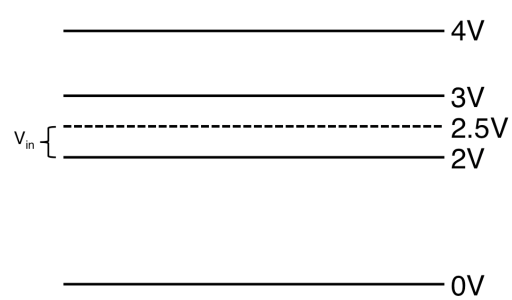
Step 2:

- Minval = 2
- Maxval = 4
- binary_representation = "1"
- Midpoint = 3
- 2.46 < 3 $\rightarrow$ binary_representation += "0"

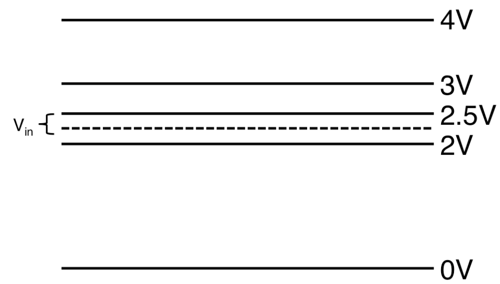


Step 3:

- Minval = 2
- Maxval = 3
- binary_representation = "10"
- Midpoint = 2.5
- 2.46 < 2.5 $\rightarrow$ binary_representation += "0"



Step 4:



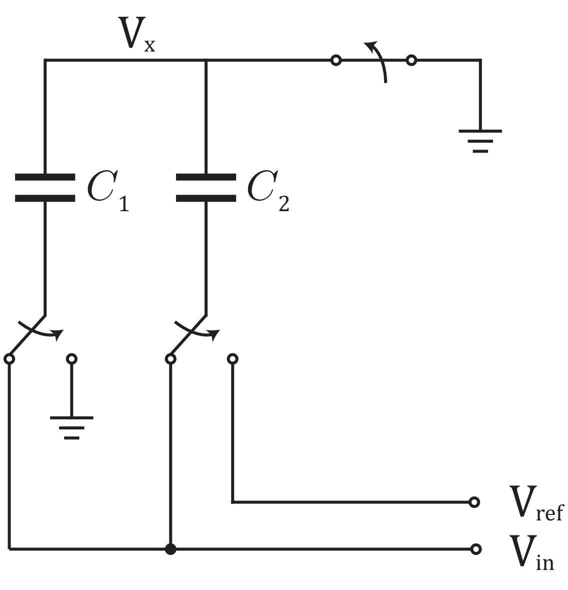
- Minval = 2
- Maxval = 2.5
- binary_representation = "100"
- Midpoint = 2.25
- 2.46 > 2.25 $\rightarrow$ binary_representation += "1"
- len(binary_representation) == 4 $\rightarrow$ return binary_representation

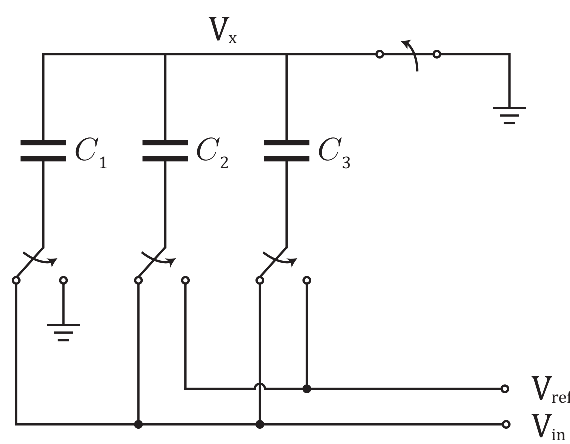
What does this binary string "1001" mean? It means V_{in} is somewhere in the range [2.25, 2.5], and the rest is up to the computer to interpret. In other words, $1001 = 1 \times \frac{V_{ref}}{2} + 0 \times \frac{V_{ref}}{4} + 0 \times \frac{V_{ref}}{8} + 1 \times \frac{V_{ref}}{16}$. This can also be written as (assuming BinaryRep is 1-indexed):

$$\text{Analog signal} = V_{in} = \sum_{i=1}^{N} V_{ref} \cdot \text{BinaryRep[i]} \cdot 2^{-i}$$

(c) **Charge Sharing**

In both of the following circuits, the switches begin in position shown in the diagram. Then, the switches change to the position indicated by the arrow (the switch connecting V_x to ground simply becomes an open circuit). First, find the total charge on the top plates of all of the capacitors before the switches transition. Then, find the voltage V_x after the switch.

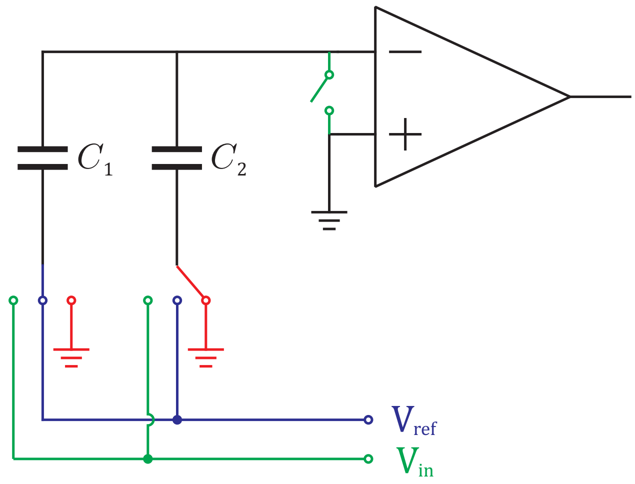




(d) **The Comparator**

A comparator is a circuit that works like an if statement. Let's suppose that in the above figures, after the switch the circuit is connected to a comparator (the triangle) like in the below diagram:

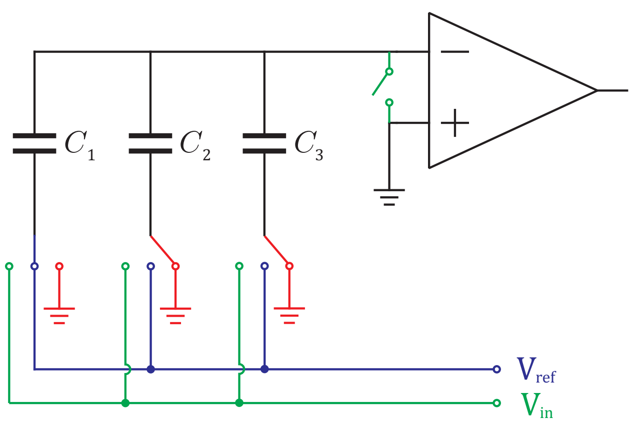
```
if(V+ >= V-):
  DIGITAL = 1
else
  DIGITAL = 0
```

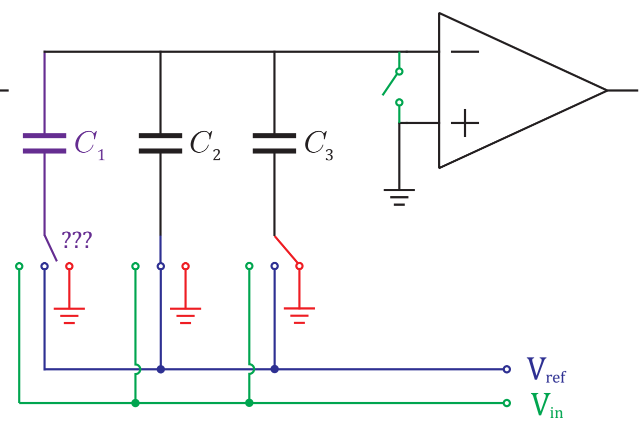


What is the comparator effectively asking? In other words, what does comparing V_x to ground imply about V_{in}, the analog signal that we care about? How does this relate to the binary search we explained before?

(e) **2-bit Successive Approximation Register Analog-to-Digital Conversion**

Now let's extend this to the next figure (left), the 2-bit version. Again, we connect the circuit (from part (c) with 3 capacitors) to the comparator. What is the comparator effectively asking? What does comparing V_x to ground imply about V_{in}?





(f) In part (e) we found the MSB of the analog signal. To find the next bit, we connect C_2's switch to V_{ref} and leave C_3 connected to ground. But we are faced with a dilemma. How do we connect C_1 (marked with ???)? The trick here is to use the result of the previous comparison. If the result of the MSB comparison was a binary 1, would we want C_1 connected to V_{ref} or to the reference voltage? What would we do if the result of the MSB comparison was 0?

(g) **The MSP430**

Generalize the 2-bit SAR ADC to 12 bits. If the smallest capacitor has a value of 1fF, what is the largest value of capacitance in the MSP430's ADC? That's how the board that you use in lab works!