Homework 2

This homework is due on Friday, February 3, 2023 at 11:59PM. Self-grades and HW Resubmissions are due the following Friday, February 10, 2023 at 11:59PM.

1. **Existence and uniqueness of solutions to differential equations**

   When doing circuits or systems analysis, we sometimes model our system via a differential equation, and would often like to solve it to get the system trajectory. To this end, we would like to verify that a solution to our differential equation exists and is unique, so that our model is physically meaningful. There is a general approach to doing this, which is demonstrated in this problem.

   We would like to show that there is a unique function $x\colon \mathbb{R} \to \mathbb{R}$ which satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = \alpha x(t) \tag{1}$$

$$x(0) = x_0. \tag{2}$$

   In order to do this, we will first verify that a solution x_d exists. To show that x_d is the unique solution, we will take an arbitrary solution y and show that $x_d(t) = y(t)$ for every t.

   (a) First, let us show that a solution to our differential equation exists. **Verify that $x_d(t) := x_0 \mathrm{e}^{\alpha t}$ satisfies eq. (1) and eq. (2).**

   (b) Now, let us show that our solution is unique. As mentioned before, suppose $y\colon \mathbb{R} \to \mathbb{R}$ also satisfies eq. (1) and eq. (2).

   We want to show that $y(t) = x_d(t)$ for all t. Our strategy is to show that $\frac{y(t)}{x_d(t)} = 1$ for all t.

   However, this particular differential equation poses a problem: if $x_0 = 0$, then $x_d(t) = 0$ for all t, so that the quotient is not well-defined. To patch this method, we would like to avoid using any function with x_0 in the denominator. One way we can do this is consider a modification of the quotient $\frac{y(t)}{x_d(t)} = \frac{y(t)}{x_0 \mathrm{e}^{\alpha t}}$; in particular, we consider the function $z(t) := \frac{y(t)}{\mathrm{e}^{\alpha t}}$.

   Show that $z(t) = x_0$ for all t, and explain why this means that $y(t) = x_d(t)$ for all t.

   (HINT: Show first that $z(0) = x_0$ and then that $\frac{\mathrm{d}}{\mathrm{d}t}z(t) = 0$. Argue that these two facts imply that $z(t) = x_0$ for all t. Then show that this implies $y(t) = x_d(t)$ for all t.)

   (HINT: Remember that we said y is any solution to eq. (1) and eq. (2), so we only know these properties of y. If you need something about y to be true, see if you can show it from eq. (1) and eq. (2).)

   (HINT: When taking $\frac{\mathrm{d}}{\mathrm{d}t}z(t)$, remember to use the quotient rule, along with what we know about y.)

2. Simple Scalar Differential Equations Driven by an Input

In this question, we will show the existence and uniqueness of solutions to differential equations with inputs. In particular, we consider the scalar differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = \lambda x(t) + bu(t) \tag{3}$$

$$x(0) = x_0 \tag{4}$$

where $u\colon \mathbb{R} \to \mathbb{R}$ is a known function of time. Feel free to assume u is "nice" in the sense that it is integrable, continuous, and differentiable with bounded derivative – basically, let u be nice enough that all the usual calculus theorems work.

(a) We will first demonstrate the existence of a solution to eqs. (3) and (4).

Define $x_d\colon \mathbb{R} \to \mathbb{R}$ by

$$x_d(t) := \mathrm{e}^{\lambda t}x_0 + \int_0^t \mathrm{e}^{\lambda(t-\tau)}bu(\tau)\,\mathrm{d}\tau\,. \tag{5}$$

Show that x_d satisfies eqs. (3) and (4).

(HINT: When showing that x_d satisfies eq. (3), one possible approach to calculate the derivative of the integral term is to use the fundamental theorem of calculus and the product rule.)

(b) Now, we will show that x_d is the unique solution to eqs. (3) and (4).

Suppose that $y\colon \mathbb{R} \to \mathbb{R}$ also satisfies eqs. (3) and (4). **Show that $y(t) = x_d(t)$ for all t.**

(HINT: This time, show that $z(t) := y(t) - x_d(t) = 0$ for all t. Do this by showing that $z(0) = 0$ and $\frac{\mathrm{d}}{\mathrm{d}t}z(t) = \lambda z(t)$, then use the uniqueness theorem for homogeneous first-order linear differential equations from the last problem. Note that the specific form of $x_d(t)$ in eq. (5) is irrelevant for the solution and should not be used.)

(c) In this part, we will calculate some values of x_d for common values of u.

i. If $u(t) := u$ is a constant function, **what is $x_d(t)$?**

ii. If $u(t) := \mathrm{e}^{\alpha t}$ for some real number $\alpha \neq \lambda$, **what is $x_d(t)$?**

iii. If $u(t) := \mathrm{e}^{\lambda t}$, **what is $x_d(t)$?**

NOTE: Assume for simplicity that $\lambda \neq 0$.

3. Hambley P4.18

Consider the circuit shown in Figure 1. Prior to $t = 0$, $v_1 = 100\,\mathrm{V}$ and $v_2 = 0$.

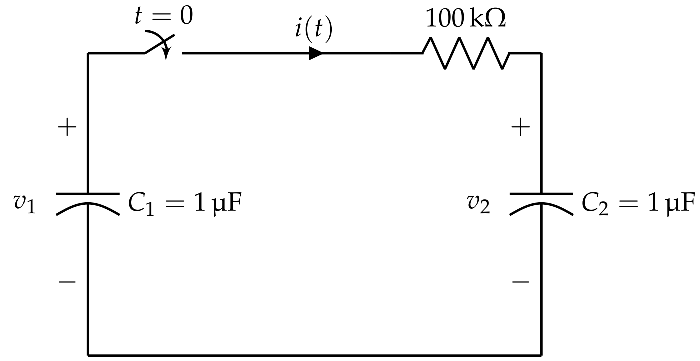


Figure 1: P4.18

(a) Immediately after the switch is closed, what is the value of the current (i.e., what is the value of $i(0+)$)?

(b) Write the KVL equation for the circuit in terms of the current and initial voltages. Take the derivative to obtain a differential equation.

(c) What is the value of the time constant in this circuit?

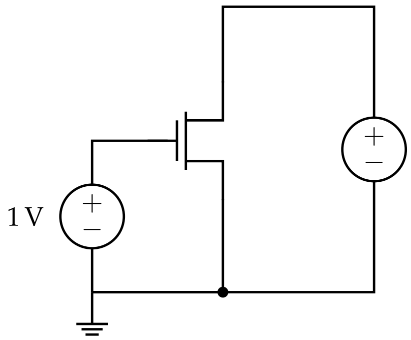
(d) Find an expression for the current as a function of time.

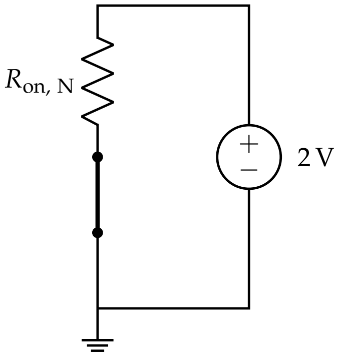
(e) Find the value that v_2 approaches as t becomes very large.

4. Transistor Behavior

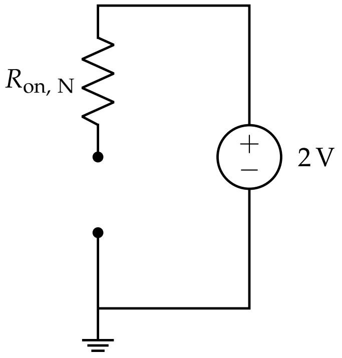
For all NMOS devices in this problem, $V_{\text{tn}} = 0.5\,\text{V}$. For all PMOS devices in this problem, $|V_{\text{tp}}| = 0.6\,\text{V}$.

(a) Which is the equivalent circuit as seen from the voltage source on the right-hand side of the circuit? **Fill in the correct bubble.**





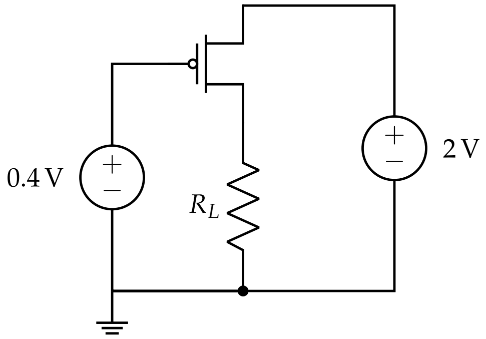
Circuit A

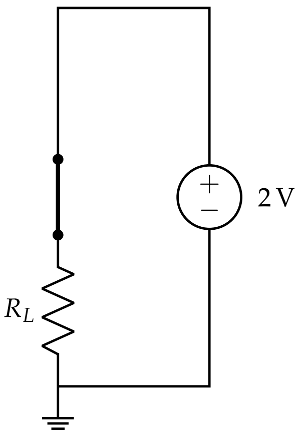


Circuit B

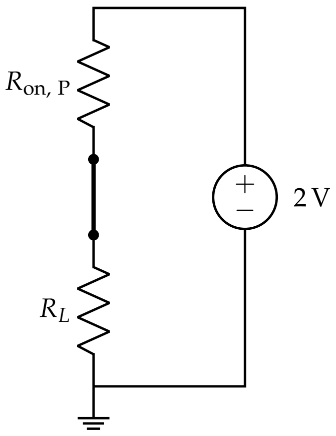
	A	B
Equivalent Circuit	○	○

(b) Which is the equivalent circuit as seen form the voltage source on the right-hand side of the circuit? **Fill in the correct bubble.**

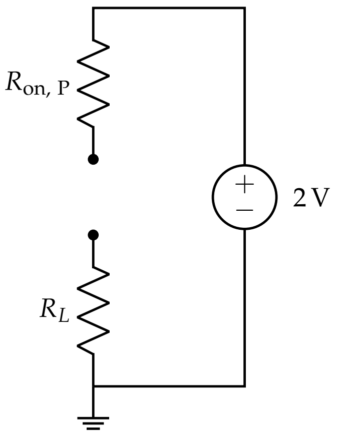




Circuit A



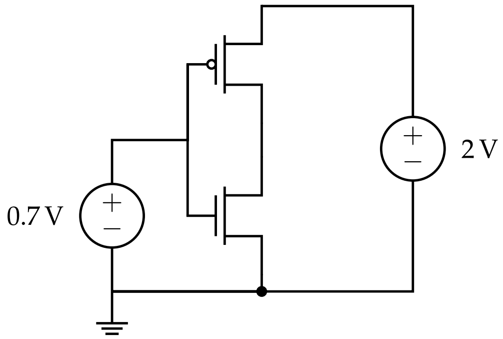
Circuit B

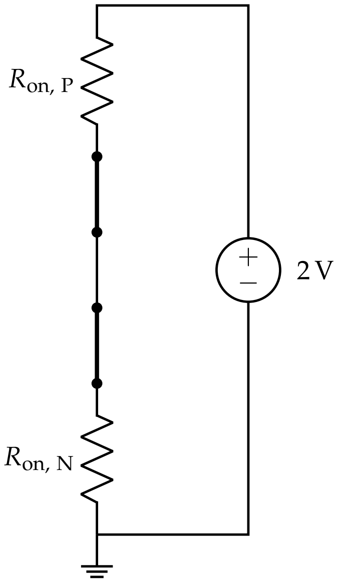


Circuit C

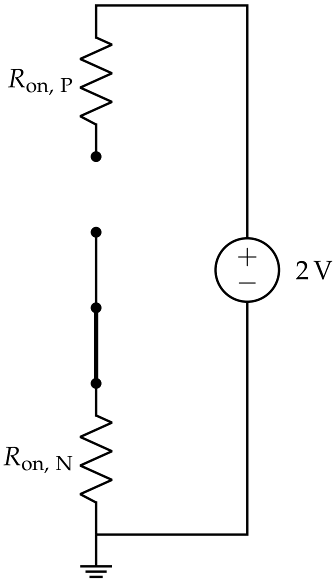
	A	**B**	**C**
Equivalent Circuit	○	○	○

(c) Which is the equivalent circuit as seen form the voltage source on the right-hand side of the circuit? **Fill in the correct bubble.**

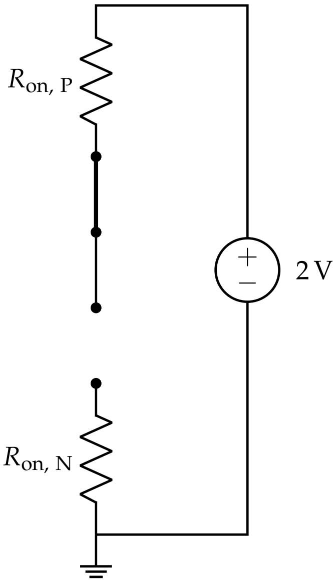




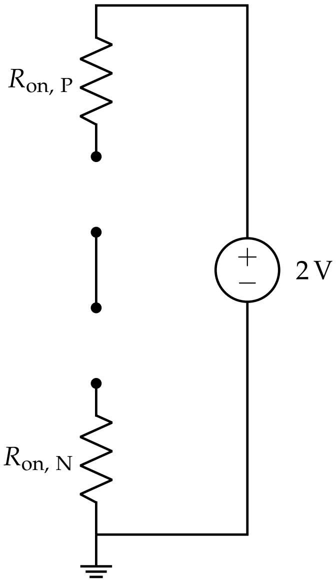
Circuit A



Circuit B



Circuit C



Circuit D

	A	B	C	D
Equivalent Circuit	○	○	○	○

5. Successive Approximation Register Analog-to-Digital Converter (SAR ADC)

An analog-to-digital converter (ADC) is a circuit for converting an analog voltage into an approximate digital representation of that voltage. One commonly used circuit architecture for analog-to-digital converters is the Successive Approximation Register ADC (SAR ADC), which you will see in Lab 3. An N-bit SAR ADC converts an input analog voltage to an N-bit binary string between 0 and $2^N - 1$. This binary string represents an integer, which in term approximates the value of our analog input voltage.

The SAR ADC does this by following one of the key themes in 16B: reducing a problem into subproblems that we already know how to solve. In this case, the two ingredients are the DAC (digital to analog converter) that we saw in HW 1, and a binary search tree that you saw in 61A. As you remember from 61A, the key operation required for a binary search is dividing a group into two halves, solving the problem at the current step with a less-than/greater-than comparison, and descending into one of the halves. We continue this until we can no longer subdivide the problem. The comparison operation is therefore key to a binary search tree. Fortunately, we have a circuit element from 16A that lets us do that: a comparator.

Explicitly, the SAR ADC implements the binary search algorithm by feeding trial digital codes into a **DAC**, like the one we analyzed in Homework 1. The circuit then takes the resuling analog voltage from the DAC and compares it with the analog input voltage using a **comparator**. It then uses feedback (**SAR logic**) to adjust the DAC output voltage to get as close as possible to the input analog voltage, step by step. The algorithm starts by determining the most significant bit (MSB), which is the bit with the largest binary weight (i.e. furthest to the left in a traditional binary number), and then moving on to determine the next bit.

If this is not clear to you, think about how you would play 20 questions to guess an integer between 0 and $2^{20} - 1$, which is approximately 1 million. This is a game where you have to guess what number your friend is thinking of, and they can only tell you if your guess is too high or too low. You have 20 guesses to get your number as close as possible to your friend's. Would you start by guessing 0, then 1, then 2, etc.? Or would you start in the middle, let's say 500,000, see if you're too high or low, then move into the next half, e.g. 250,000 or 750,000 depending on your result? The latter approach of divide-and-conquer is a faster way of solving the problem! The SAR ADC is basically a circuit implementation of this game. The input voltage is the "number" your friend has in mind, the DAC code is your guess, the comparator is your friend's response (too high vs. too low), and the SAR logic used to adjust the next code is you deciding what to guess next. For a 20-bit SAR DAC, you have 20 guesses; for an N-bit SAR DAC, you have N guesses.

Figure 8 illustrates a high-level circuit diagram of a SAR ADC. The analog input voltage is one of the inputs to the comparator. The other comparator input, V_{DAC}, is the DAC voltage output, i.e. the analog representation of your code/guess. The comparator compares these two values and outputs a logical high (1) if $V_{IN} \geq V_{DAC}$ or a logical low (0) if $V_{IN} < V_{DAC}$. This comparator decision then feeds into the SAR logic, implemented in a microcontroller in this case. This logic is basically the brain that implements the binary search pattern, deciding which bits in the code need to be changed. The updated code then propagates back to the DAC, which updates V_{DAC}, and the cycle continues N times. Once this is done, the final N-bit output code comes out of the register storage units for use by other circuits. If you're wondering what V_{REF} does for the DAC, recall from Homework 1 that it tells the DAC what the maximum possible input voltage is.

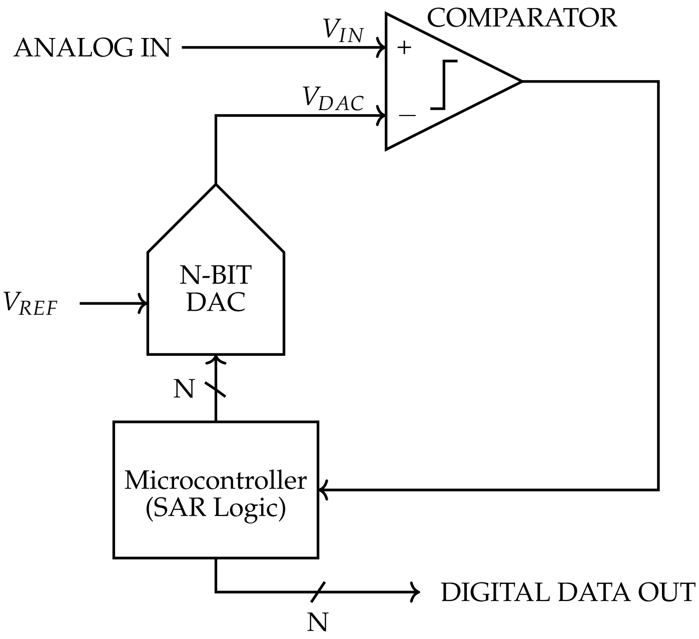


Figure 8: SAR ADC circuit diagram

Here is an illustration of the algorithm, which the SAR logic takes care of.

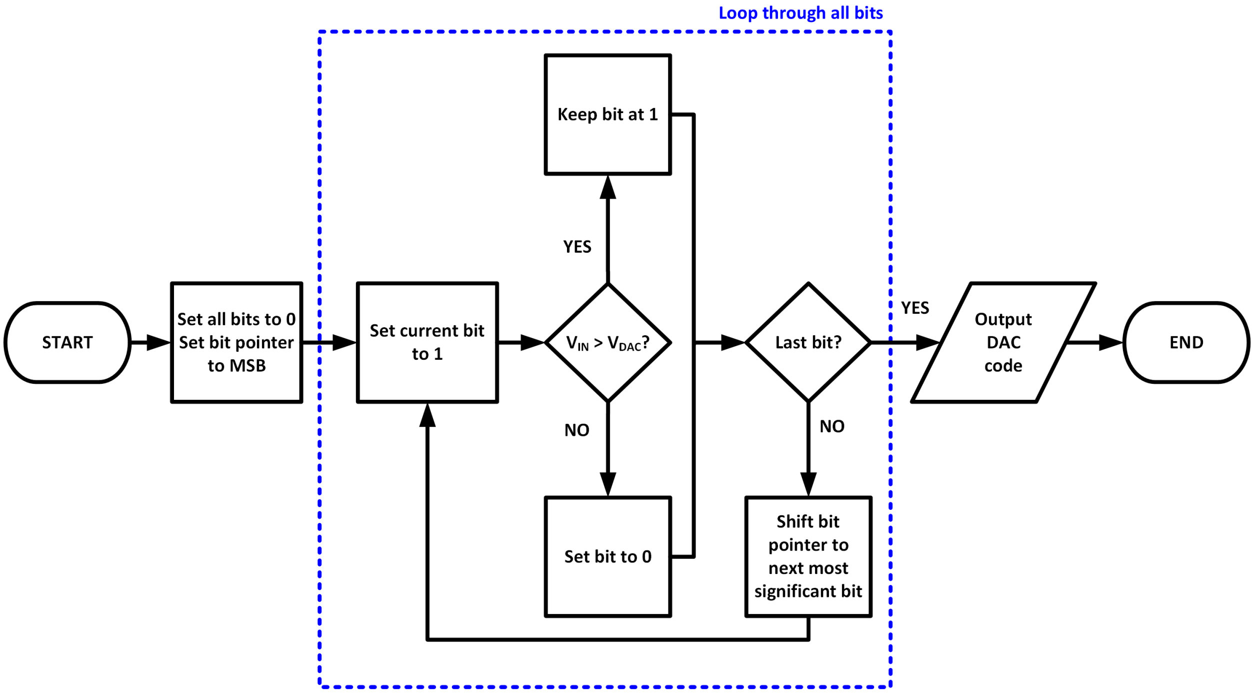


Figure 9: Flow chart of SAR ADC algorithm.

Let's look at a 3-bit SAR ADC as an example. The voltage transfer curve of the 3-bit SAR ADC is shown in Figure 10. For $V_{\text{IN}} = 0.3V_{\text{REF}}$, the operation of SAR ADC is shown in Figure 11.

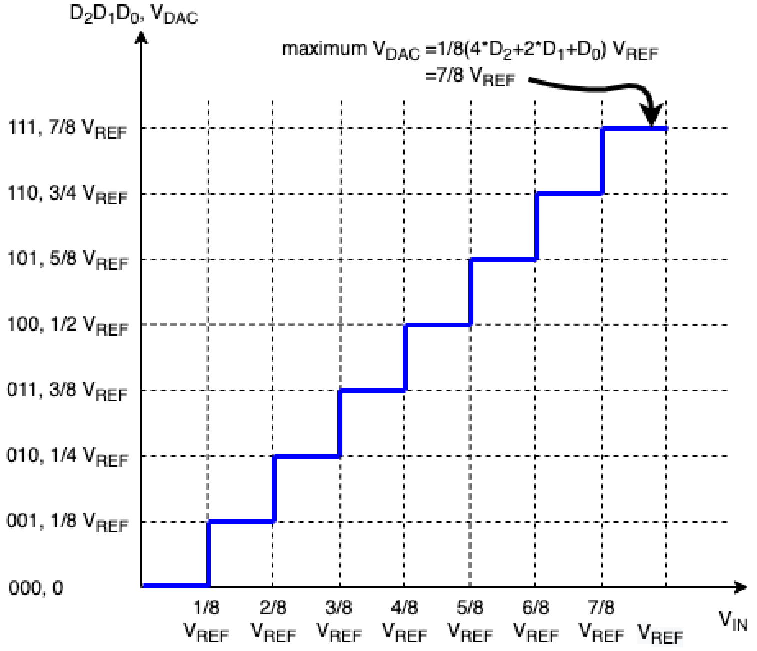


Figure 10: Voltage transfer curve of 3-bit SAR ADC: $D_2D_1D_0$ is the output digital code of the ADC. Notice that the maximum voltage of the V_{DAC} is $\frac{2^N-1}{2^N}V_{\text{REF}}$ where N is the number of bits ($N = 3$ in this case)

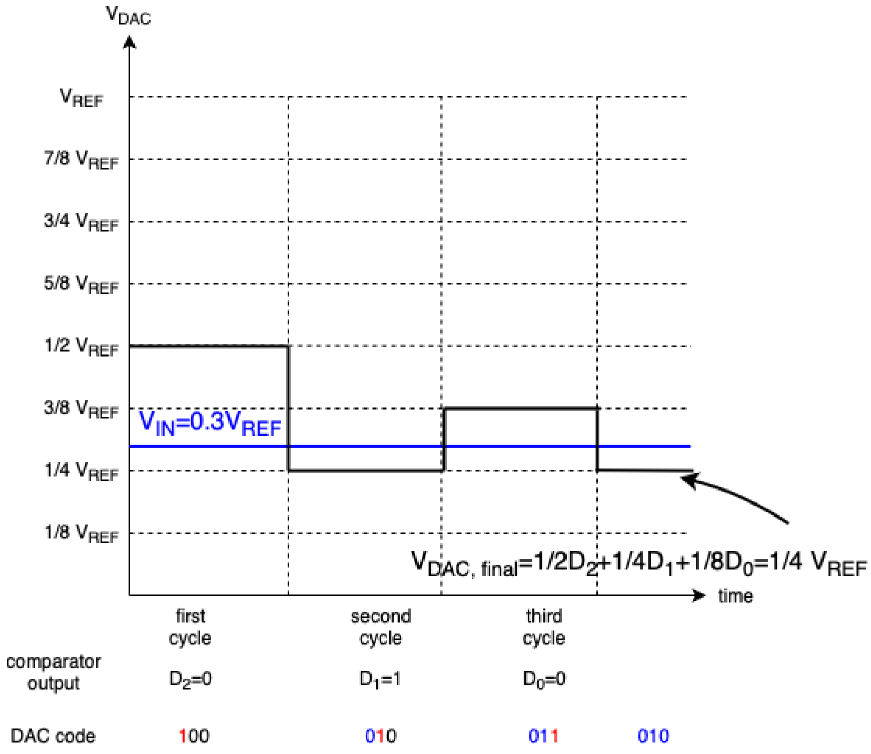


Figure 11: Timing diagram of SAR output code. Red digit in DAC code is the bit being decided in the cycle; blue digits are the bits determined by the previous conversion cycles.

From the timing diagram in Figure 11, it can be seen that the final digital output code is 010 which represents the closest DAC output voltage $\left(\frac{1}{4}V_{\text{REF}}\right)$ to the input analog voltage $V_{\text{IN}} = 0.3V_{\text{REF}}$ with a 3 bit binary representation.

We will now analyze the operation of a 4-bit SAR ADC with the input voltage range between 0 V and 5 V and output code range between 0000 (0) and 1111 (15). In other words, V_{REF} is 5 V and $0\,\text{V} < V_{\text{in}} < 5\,\text{V}$ in Figure 8.

(a) **If the ADC output code is 0000, what is the corresponding DAC voltage? What about code 1111? Code 1001?** *(HINT: Try to draw the transfer curve for 4-bit SAR ADC, similar to that for the 3-bit SAR ADC in Figure 10.)*

(b) **If the input analog voltage is $V_{\text{IN}} = 3\,\text{V}$, what is the ratio between the input voltage V_{IN} and the maximum input voltage $V_{\text{REF}} = 5\,\text{V}$? What should be the output code of the SAR ADC?**

(c) Again, if the input analog voltage is $V_{\text{IN}} = 3\,\text{V}$, draw the operation of a 4-bit SAR ADC resolving its output code (see Figure 11 as reference). Specifically:

   i. **Plot the output voltage of the DAC in the timing diagram in Figure 12.**

   ii. **Fill out the output of the comparator in each conversion cycle.**

   iii. **Fill out the ADC output code ($D_3D_2D_1D_0$) in each conversion cycle.**

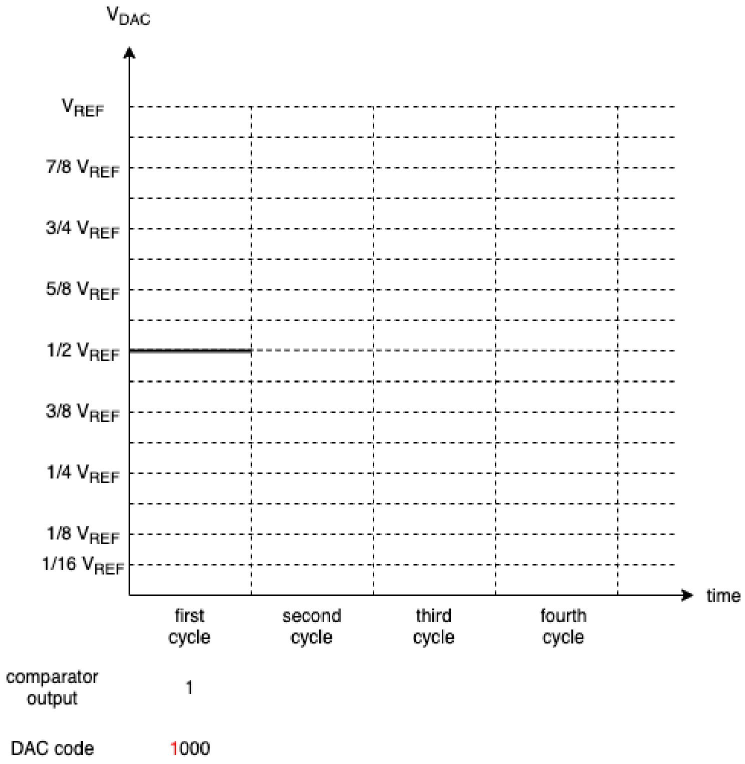


Figure 12: SAR ADC V_{DAC} timing diagram

Contributors:

- Anant Sahai.
- Nikhil Shinde.

- Druv Pai.
- Sally Hui.
- Sidney Buchbinder.
- Yi-Hsuan Shih.
- Vladimir Stojanovic.
- Ayan Biswas.
- Wahid Rahman.