Homework 2

This homework is due on Saturday, February 3, 2023 at 11:59PM.

The homework relies on material covered in lecture on inductors (**01/23**) and transistors (**01/25**) as well as the corresponding notes, **Note 3** and **Note 4** respectively.

1. Inductor Properties

The current in a 100 mH inductance is given by $0.5\sin(1000t)$A. Find expressions and sketch the waveforms to scale for the voltage, power, and stored energy, allowing t to range from 0 to 3π ms. The argument of the sine function is in radians.

2. Mutual Inductance

The pair of mutually coupled inductances in Figure 1 has $L_1 = 2\,\text{H}$, $L_2 = 1\,\text{H}$, $i_1 = 2\cos(1000t)\text{A}$, $i_2 = 0$, and $v_2 = 2000\sin(1000t)\text{V}$. (The dots in the figure indicate orientation and the arguments of the sine and cosine functions are in radians.) Find $v_1(t)$ and the magnitude of the mutual inductance.

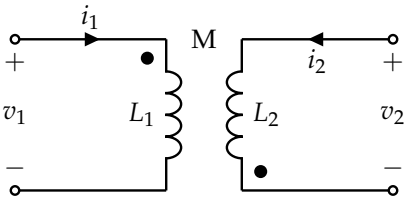


Figure 1: Circuit diagram.

3. Steady-State Circuit

Solve for the **steady-state** values of i_1, i_2, and i_3 in the circuit shown in Figure 2.

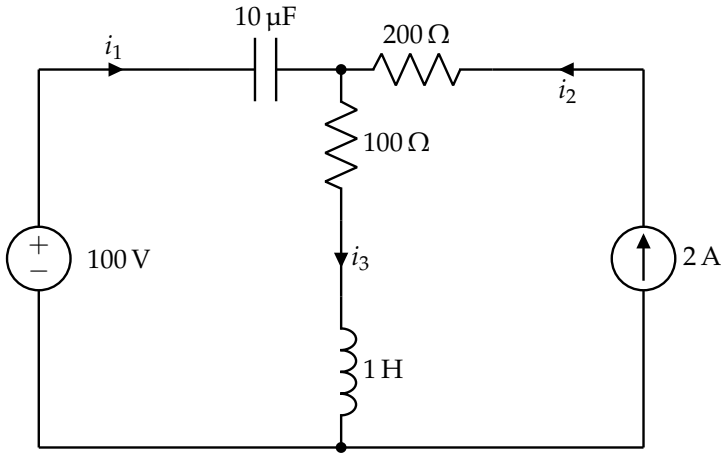
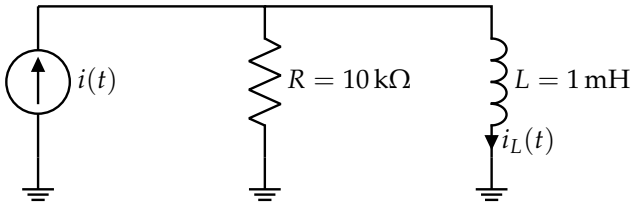


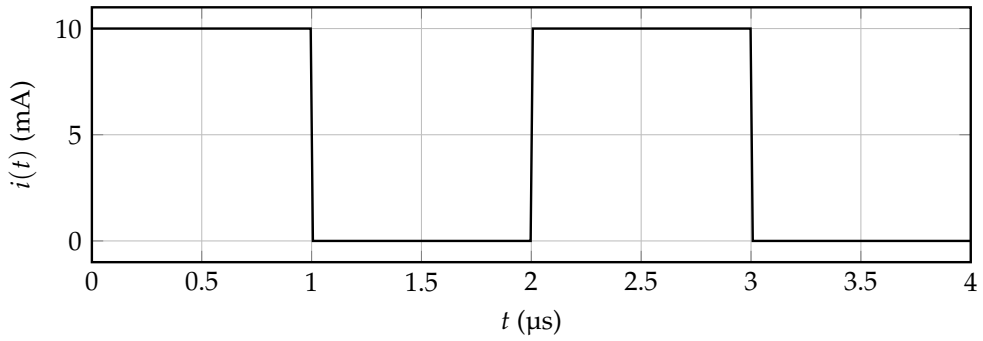
Figure 2: Circuit diagram.

4. RL Circuit with Square Wave Input

In this problem, we will explore how an RL circuit behaves to a square wave input.

Suppose you have the following circuit, with the plot of input $i(t)$ shown below. Assume $i_L(0) = 0$.

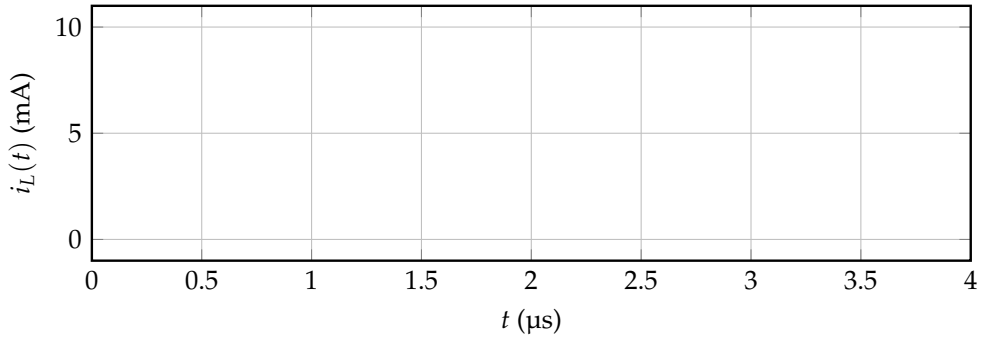




(a) Find the **differential equation for** $i_L(t)$ in terms of R, L, and $i(t)$.

(b) Find the **time constant** τ that describes this circuit both in terms of R and L, as well as numerically using the element values provided in the circuit diagram.

(c) **Qualitatively draw $i_L(t)$ for $0\,\mu s < t < 4\,\mu s$** on the provided plot (make sure to look at the units of the plot). Also, remember that $i_L(0) = 0$.



(HINT: Remember how the time constant relates to transitions between voltages.)

For the rest of the problem, we will focus on the time interval $0\,\mu s < t < 1\,\mu s$ (which represents the first transition for the circuit).

(d) **Solve the differential equation for $i_L(t)$** on the time interval $0\,\mu s < t < 1\,\mu s$.

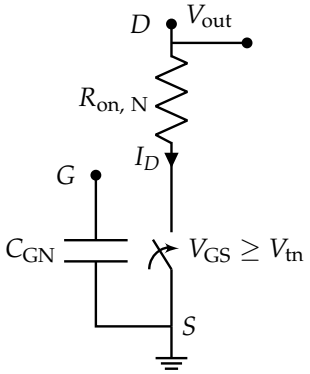
(e) **(Optional)** Suppose we want to shorten the time period of the input square wave $i(t)$ such that the first transition occurs over the time interval $0 < t < T$ (currently, $T = 1 \times 10^{-6}\,s = 1\,\mu s$). However, we want to still ensure that $i_L(t)$ reaches at least 90% of $i_{L,\text{desired}}$, its target/desired value (the value it approaches over the time interval). To find the minimum time T such that this condition is fulfilled, we can solve the following equation:

$$\frac{i_L(T)}{i_{L,\text{desired}}} = 0.9 \tag{1}$$

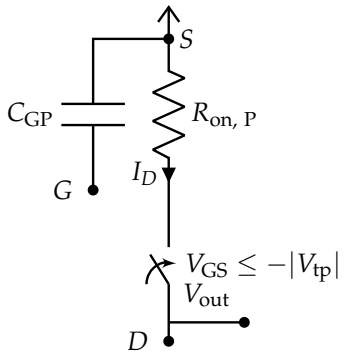
Using your answer from the previous part, **solve for the minimum time** T. Your answer should be numerical, but you do not need to simplify it.

5. Transistor Switch Model

We can improve our resistor-switch model of the transistor by adding in a gate capacitance. In this model, the gate capacitances C_{GN} and C_{GP} represent the lumped physical capacitance present on the gate node of all transistor devices. This capacitance is important as it determines the delay of a transistor logic chain.



(a) NMOS Transistor Resistor-switch-capacitor model



(b) PMOS Transistor Resistor-switch-capacitor model. Note we have drawn this so that it aligns with the inverter.

You have two CMOS inverters made from NMOS and PMOS devices. Both NMOS and PMOS devices have an "on resistance" of $R_{on,\,N} = R_{on,\,P} = 1\,\text{k}\Omega$, and each has a gate capacitance (input capacitance) of $C_{GN} = C_{GP} = 1\,\text{fF}$ (fF = femto-Farads = 1×10^{-15} F). We assume the "off resistance" (the resistance when the transistor is off) is infinite (i.e., the transistor acts as an open circuit when off). The supply voltage V_{DD} is 1V. Assume $V_{DD} > V_{tn}, |V_{tp}| > 0$. The two inverters are connected in series, with the output of the first inverter driving the input of the second inverter (Figure 5).

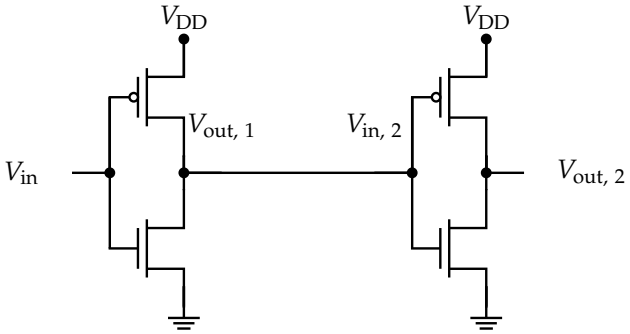


Figure 4: CMOS Inverter chain

(a) Assume the input to the first inverter has been low ($V_{in} = 0$ V) for a long time, and then switches at time $t = 0$ to high ($V_{in} = V_{DD}$).

Draw a simple RC circuit and write a differential equation describing the output voltage of the first inverter ($V_{out,\,1}$) for time $t \geq 0$.

Don't forget that the second inverter is "loading" the output of the first inverter — you need to think about both of them.

(HINT: Your simple RC circuit model will only have 3 elements; you only need to draw the elements that impact the behavior of $V_{out,\,1}$ and thus are relevant in this specific scenario. Also, for the first inverter,

when $V_{\text{in}} = V_{\text{DD}}$, the NMOS transistor model's switch will be closed while the PMOS transistor model's switch will be open.)

(b) **Solve for $V_{\text{out, 1}}(t)$.** The initial condition will be $V_{\text{out,1}}(0) = V_{DD}$ (this can be found by using the situation described in part (a)).

(c) **Sketch the output voltage of the first inverter, showing clearly (1) the initial value, (2) the asymptotic value, and (3) the time that it takes for the voltage to decay to roughly 1/3 of its initial value.** *(HINT: For part (3), use the approximation that $\text{e}^{-1} = \frac{1}{\text{e}} \approx \frac{1}{3}$.)*

6. (Lab) 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 5 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.

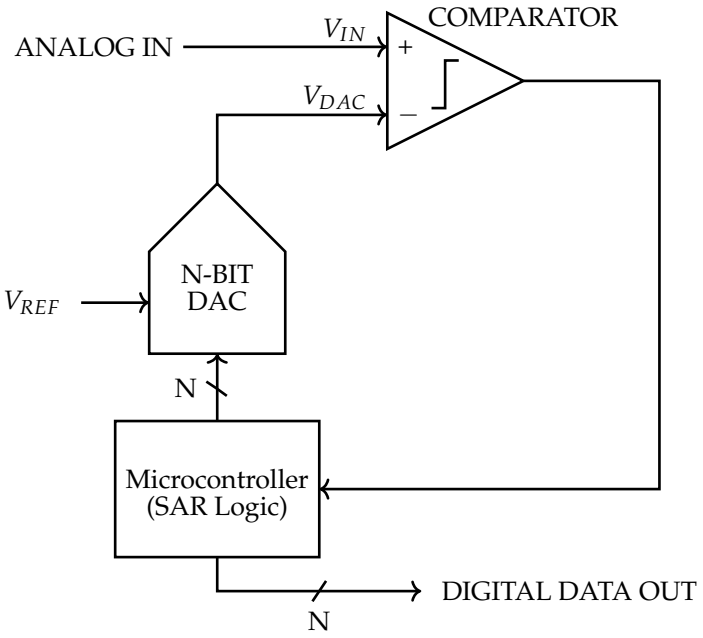


Figure 5: SAR ADC circuit diagram

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

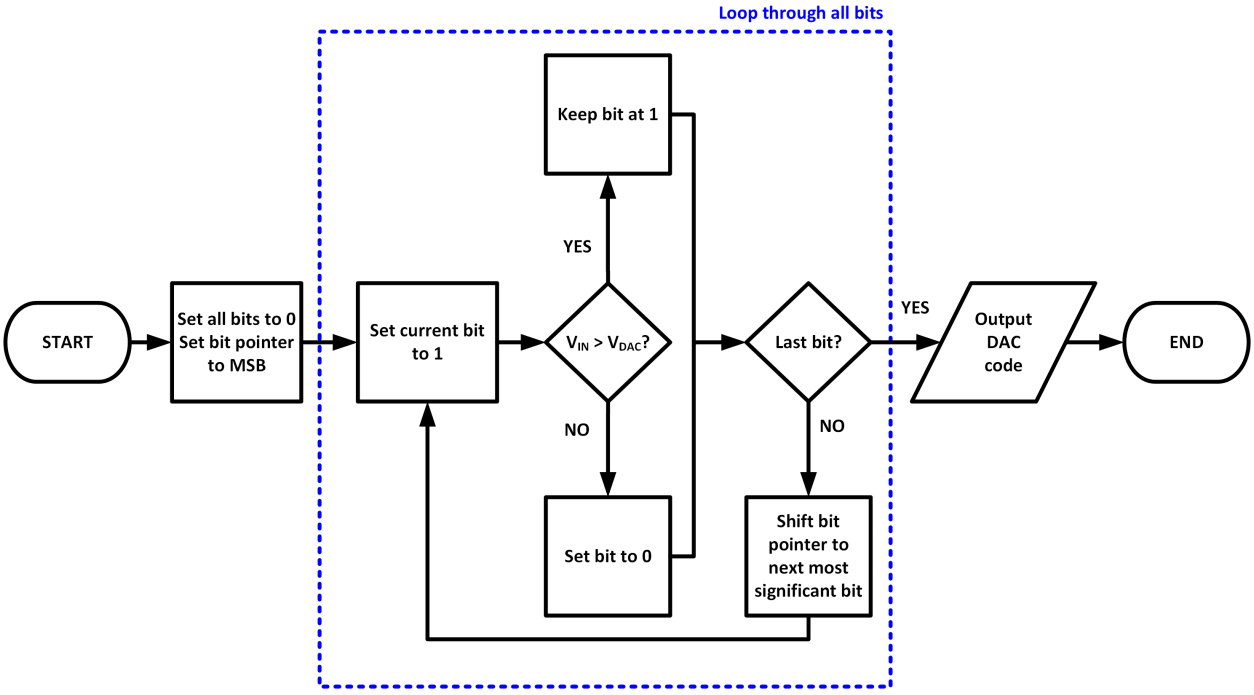


Figure 6: Flow chart of SAR ADC algorithm.

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 7. For $V_{\text{IN}} = 0.3V_{\text{REF}}$, the operation of SAR ADC is shown in Figure 8.

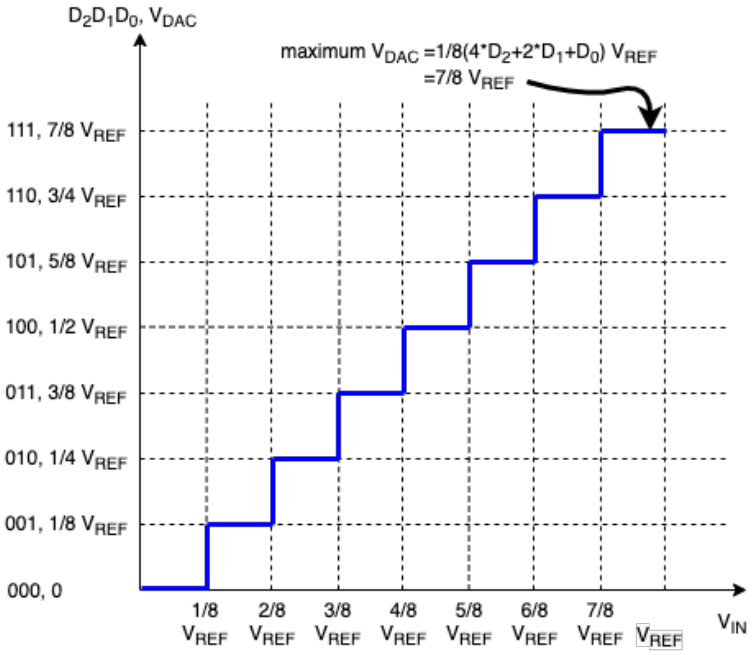


Figure 7: Voltage transfer curve of 3-bit SAR ADC: $D_2D_1D_0$ is the output digital code of the ADC. Notice that the maximum voltage of the V_{DAC} is $\frac{2^N-1}{2^N}V_{\text{REF}}$ where N is the number of bits ($N = 3$ in this case)

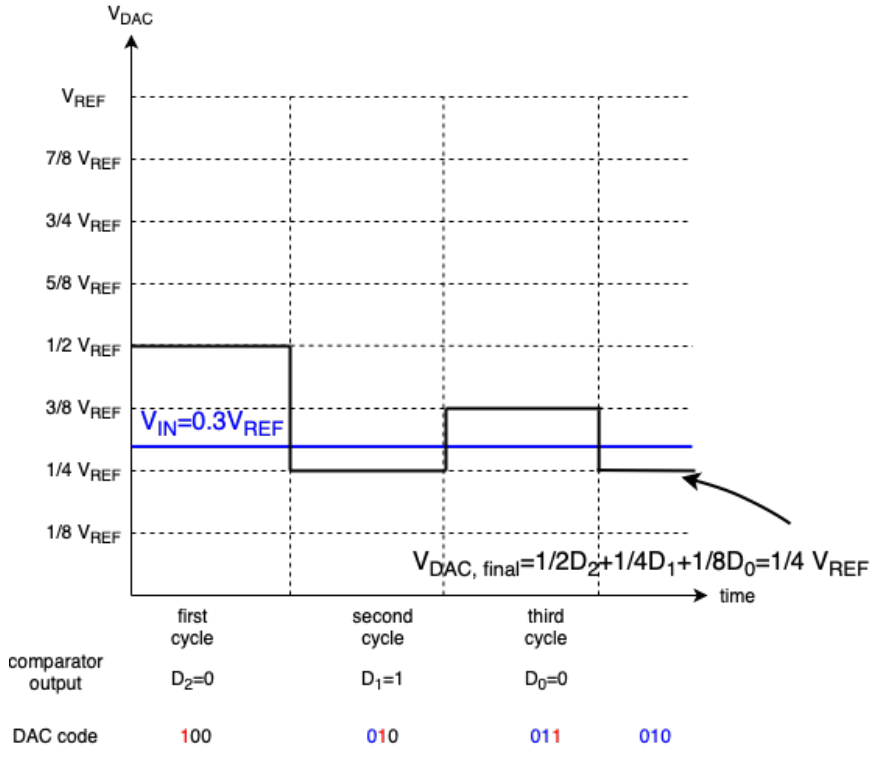


Figure 8: Timing diagram of SAR output code. Red digit in DAC code is the bit being decided in the cycle; blue digits are the bits determined by the previous conversion cycles.

From the timing diagram in Figure 8, 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 5.

(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 7.)*

(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 8 as reference). Specifically:

i. **Plot the output voltage of the DAC in the timing diagram in Figure 9.**

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.**

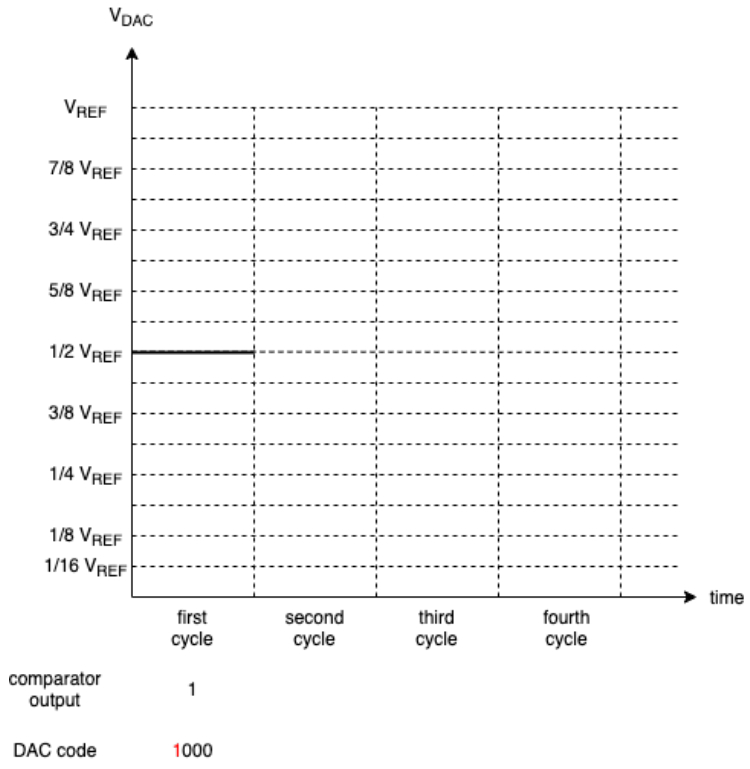


Figure 9: SAR ADC V_{DAC} timing diagram

Contributors:
- Kris Pister.
- Regina Eckert.

- Sidney Buchbinder.
- Ayan Biswas.
- Gaoyue Zhou.
- Wahid Rahman.
- Nikhil Jain.
- Chancharik Mitra.
- Yi-Hsuan Shih.
- Vladimir Stojanovic.
- Allan Hambley.