Design of Integrated Temperature Sensor with 3-bit digital output over the range $-13.5°C$ to $70.5°C$

Brandon Lawrence, *Student, NTNU,* Mathis Bonnard, *Student, NTNU*

***Abstract*—In this paper, we propose a design of an integrated temperature sensor with a digital output. We use a cricuit to have a current proportional to the temperature, using two PMOS, two PNP and a custom Operational Amplifier. The analog-to-digital converter is a flash ADC with some logic to transform a thermometer code into a 3-bit output. The whole design is made to fit in a chip manufactured using the SKY130 process node.**

***Index Terms*—Temperature sensor, Flash ADC, Current proportional to temperature, SKY130.**

I. Introduction

In THE course TFE4188 – Advanced Integrated Circuits at NTNU, our final project is to *Design an integrated temperature sensor with digital read-out*. This temperature sensor is designed using the SKY130 process node.

Temperature sensors are commonly used in most chips to ensure that the heat generated by normal operation is not excessive and doesn't affect the normal functioning. Having a digital read-out of such sensor is then very valuable as it can be used as input to the control unit. Temperature sensors can also be used to monitor ambient temperature which has many uses.

This circuit then needs two different parts: a temperature-linked quantity (current or voltage) and an Analog-to-Digital Converter (ADC).

To have the temperature-linked quantity, we decided to use a current-proportional-to-temperature (PTAT) circuit [1]. Then, we link the output of this circuit to a flash ADC [2] and a logic decoder to have a 3-bits digital output. This architecture does not provide a highly-precise result, but we have a parallel output with virtually zero delay between a change in temperature to the change in output.

April 2023

II. Theory

As we want to measure a temperature which is a continuous value quantity, with no obvious link to electricity, our design needs at least a temperature sensor and an Analog-to-Digital Converter (ADC).

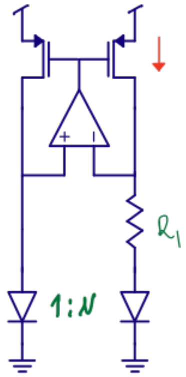

Fig. 1. Circuit providing a current proportional to temperature. [1]

First, the temperature sensor must have an output that varies monotonically with temperature and this output has to be an electric value such as voltage or current. Ideally, this output should be related to the temperature by a linear function. This means that we need a circuit that ”translates” temperature into voltage or current with a well-defined linear relationship.

After creating this well-defined signal, the ADC converts the continuous input to a digital read-out. Two different quantities can be used to characterise the ADC: the number of output bits and the output refresh rate. Ideally these parameters should be high to increase precision and minimise delay of the output.

A. Current Proportional to Temperature

We want to obtain a tension proportional to temperature. This tension can be a voltage drop across a resistor reflecting a current proportional to temperature.

The circuit shown in Fig. 1 provides a voltage linear to the temperature across the resistor R_1. This can be shown using the characteristics of an ideal Operational Amplifier (OpAmp) and the diodes and CMOS equations. Let D_1 be the left-most diode and D_2 be the right one.

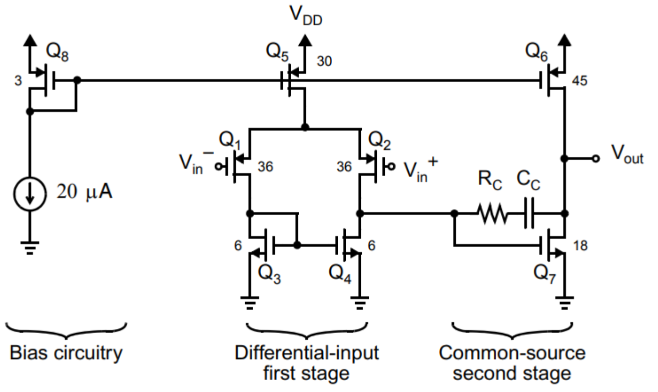

Fig. 2. A CMOS two-stage amplifier. All transistor lengths are 0.3 μm and widths are shown in μm next to each transistor. [2]

The forward-biased diode equation [2] describes that the voltage V_{Di} across the diode D_i is

$$V_{Di} = V_T \ln \frac{I_{Di}}{I_S}$$

Where I_S is a constant, $I_{D1} = NI_{D2}$ because of the size ratio and

$$V_T = \frac{kT}{q} \propto T$$

The OpAmp will force the voltage on top of the resistor to be equal to V_{D1}, thus the voltage ΔV_D across the resistor R_1 is

$$\Delta V_D = V_T \ln N$$

Then, it is worth noting that with T being the absolute temperature, we have:

$$\Delta V_D = \frac{kT}{q} \ln N \propto T$$

This circuit should then give a tension proportional to the absolute temperature if the OpAmp is ideal enough to assume that $V^+ = V^-$.

B. Operational Amplifier

The circuit in Fig. 1 needs a well-designed OpAmp to work as expected. The OpAmp must have good characteristics to keep its input to the same voltage. Then, we need quite a high gain for the design of our OpAmp.

The book [2] gives the design of a two-stage amplifier, as shown in Fig. 2. The gain A_V of this design of two-stage OpAmp can be approximated by the product of the gain of each stage:

$$A_V = g_{m1} g_{m7} \left(r_{ds2} || r_{ds4}\right) \left(r_{ds6} || r_{ds7}\right)$$

This design should provide an OpAmp with sufficient characteristics to be used inside the temperature sensor.

C. Analog-to-Digital Converter

The design uses a Flash ADC, this is a very simple ADC as shown in Fig. 5 of Appendix A. The ADC consists of a voltage ladder as in Fig. 6 connecting a reference voltage to a comparator between each resistor.

The other input to each comparator is provided by the continuous signal proportional to temperature. Thus the output of each comparator should successively increase with a steep change from $0V$ to VDD, or logic low to high as input voltage increases, this provides a useful output that can be converted to a digital output by means of an encoder.

Connecting the Voltage Ladder to the temperature dependent circuit required an additional two resistors that can be seen in the final design schematic Fig. 12, these allowed for the outputs to be tuned to specific input voltage and reference voltage conditions.

D. Using OpAmp as Comparator

To implement the comparators in the ADC, simply using an OpAmp with high gain was sufficient, providing an accurate output with suitable speed as seen in Fig. 13. The schematic of the OpAmp in Fig. 7 operates in the same way as the differential-input first stage sub-circuit of Fig. 2 and is outlined in the textbook [2], as such the gain equation for A_V is given as:

$$A_V = g_{m5}(r_{ds1} || r_{ds6})$$

To achieve the results in Fig. 13 the parameters of PMOS transistors are $L = 0.3\mu\text{m}$, $W = 36\mu\text{m}$, and NMOS are $L = 0.3\mu\text{m}$, $W = 6\mu\text{m}$.

E. Output Logic

After receiving the output of the voltage ladder, the encoder outputs a 4-bit binary output as shown in Table II using digital logic.

III. Implementation

How we did. Tools, simulations parameters

A. Tools

As our project is open-source, we only used open-source tools to design and simulate the entire temperature sensor. As such, the design of circuits was made using xschem[1] and we simulated everything with ngspice[2]. Cicconf[3] and cicsim[4] were also required to automate the ngspice simulations.

[1] https://xschem.sourceforge.io/

[2] https://ngspice.sourceforge.io/

[3] https://github.com/wulffern/cicconf

[4] https://github.com/wulffern/cicsim

B. Design process

The final design of our temperature sensor is the result of several modifications of an expected-to-work design. We first drew the architecture of the sensor with the technology we aimed to use for each part. Then, we designed each main section before linking them together. We split the whole project into two main parts: a temperature sensor with an analog output and an Analog-to-Digital Converter connected to an encoder and binary output.

When both parts were designed, they were simulated separately. Simulation results led us to modify some of our design characteristics until satisfactory results were obtained. Then, we linked them to check that the overall temperature to output voltage was as expected.

1) Temperature sensor and OpAmp: The theory gives a voltage drop proportional to the temperature across the resistor [1]. But the voltage both at the top and the bottom of the resistor varies with temperature. Then, to link the flash ADC to the tension, we decided to not use the voltage across the resistor, but only the voltage at the top of this one. Hence, the ADC only had to be calibrated for the highest and the lowest tension at this point to have a digital output linked to the temperature. This means that the highest voltage value was given at low temperature and vice versa.

In order to have acceptable results with a Monte Carlo simulation (see III-C1), we chose transistors with a large area (around $100\mu m^2$). The ratio W/L of the transistors where given by the book [2] for the OpAmp and we choose $W = 10$ and $L = 1$ for the two PMOS. This allowed enough current to flow through the resistor give a large enough voltage drop.

We decided to replace the diodes by bipolar transistors in their active region. The ratio between both transistors is $1 : 8$ because $\ln(8) \approx 2$ and then the current in the rightmost transistor is twice the current in the leftmost one (see II-A).

The implementation of the PTAT circuit is shown in the Fig. 4 in the Appendix B.

2) ADC: The ADC is based on a voltage ladder as reference. This ladder was manually tuned by altering the resistor values to be in sync with temperature sensor. Hence, the resistor between the VSS line and the first comparator set the voltage reference for this comparator at the tension reached for $T = 70.5°C$ at the top of the resistor. Similarly, the resistor between the last ADC and the VDD line set the voltage reference at the tension reached for $T = -13.5°C$ at the top of the resistor. Then, the intermediate resistors are of equal resistance to change output logic at uniform increments.

3) Output logic: Due to the successive increase in logic highs, of the voltage ladder output, the digital logic is simple to implement. Using the second row of Fig. II as an example, the output of 'a' was simply "a AND NOT b", similarly the fourth row logic was "c AND NOT d", this method can be followed for all outputs and was implemented in Fig. 11. The digital logic gates were also implemented using CMOS design, schematics for an inverter, NAND, and OR gate are shown in Fig. 10, Fig. 9, and Fig. 8 respectively. The outputs of logic gates were as expected so have been omitted here.

C. Simulations parameters

1) Temperature sensor: To have a good idea of the actual behaviour of our design in real conditions, we simulated the temperature sensor at the typical corner and in extreme conditions. We also added a Monte-Carlo simulation to handle the mismatch between theoretically same devices. These simulations were repeated for different temperatures to check for any non-linear phenomena.

We simulated our design this way to have an overview of the behaviour of the design of our temperature sensor across all corners and for different manufacturing mismatches at a wide range of temperatures. The results presented in this paper are the simulations for the final design.

2) ADC and encoder: We simulated the ADC and the encoder using the typical corner. We believe this sub-circuit to be more likely to work across all corner and as simulations take a significant amount of time. Then, we chose to assume that as long as the design has the expected behaviour for the typical corner, the design should work at all corners. This assumption was supported in the process of joining the two circuits, with results that agreed with this assumption.

IV. Results

A. Temperature sensor

We characterised the temperature sensor by doing simulations at different corners. We do the typical corner, which will be considered as the reference value. We also check the extreme corners to ensure the design would work correctly in production. We also check the influence of the mismatch using a Monte Carlo method to have an idea of how many devices would be considered as non-functional after production.

The only variable we varied between simulations was the temperature. We simulated the design for $T = -25,\ 0,\ 27,\ 50,\ 75$ and $100°C$. The Table IV in the Appendix B compiles the voltage drop across the resistor according to the simulations. The leftmost "Typ" is for

TABLE I
OUTPUT TENSION (IN mV) AT THE TOP OF THE RESISTOR FOR ALL CORNER AT DIFFERENT TEMPERATURE

T (°C)	Typ	Min	Typ	Max	-3std	mean	+3std
-25	867	861	868	875	852	867	882
0	835	832	838	846	821	837	853
27	798	793	802	811	783	796	810
50	765	760	769	780	750	763	776
75	729	723	734	746	714	730	745
100	693	686	699	712	677	695	713

TABLE II
DIGITAL LOGIC OF BINARY ENCODER

Thermometer Input							Binary Output			
g	f	e	d	c	b	a	d	c	b	a
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	1	0	0	1	0
0	0	0	0	1	1	1	0	0	1	1
0	0	0	1	1	1	1	0	1	0	0
0	0	1	1	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1

the standard typical condition. The ”Min” is for the slowest extreme corners while ”Max” is for the fastest and the rightmost ”Typ” is for the average. The columns ”-3std” and ”+3std” are estimations of the results for a distance of 3 sigmas around the mean in the Monte Carlo simulation.

For the typical corner, the linear regression of the voltage drop across the resistor as a function of the temperature is

$$V_{out} = 0,2453T + 54.866$$

(voltage drop in mV and temperature in $°C$) with a correlation coefficient $R^2 = 0.999$.

Similarly, the tensions at the top of the resistor are given in the Table I.

For the typical corner, the linear regression of the tension at the top of the resistor as a function of the temperature is

$$V_{out} = -1.3997T + 834.23$$

(tension in mV and temperature in $°C$) with a correlation coefficient $R^2 = 0.9996$.

The sweep of the tension at the top of the resistor is higher than the sweep of the voltage drop across it. Then, it’s easier to build a precise ADC that takes an input of the tension at the top of the resistor rather than the voltage drop. For the same temperature difference, the voltage difference is larger in the first case.

B. ADC

To test the ADC, we split it into two sub-circuits of the Voltage Ladder and Digital Encoder. Fig. 3 shows the outputs of each comparator of the Voltage Ladder when VDD $= 1.8V$, and reference voltage is $0.9V$. These outputs correlate to the thermometer input of Table II.

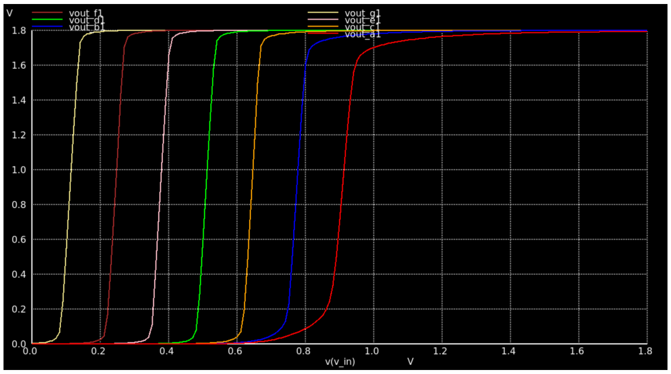

Fig. 3. Outputs of Voltage ladder with $V_{ref} = 0.9V$.

The digital encoder was simple to implement and operated as expected, the output values were virtually identical to Fig. II where logic high was 1.8V.

It was determined manually that the ADC remained functional at low values of VDD, with minimal skew in the expected output values. However at low-voltage, high-temperature conditions the precision of the encoder was compromised and so we decided to maintain VDD at 1.8V and only alter reference voltage.

C. Temperature sensor with digital read-out

As the tension at the bottom of the resistor changes with temperature, linking the temperature sensor with the flash ADC caused problems for the reference voltage (voltage at which the lower comparator should trigger and turn to 1). Considering also the voltage range, we decided to use only the tension at the top of the resistor of Fig. 1 as input for the ADC (see Table I in the Appendix B).

After implementing this and joining all sub-circuits, the whole design was simulated at various temperatures to obtain a digital output as shown in Table III.

TABLE III
BINARY OUTPUT AT TEMPERATURE RANGES

Temperature range (°C)	Binary Output		
	A	B	C
$T <$ -13.5	0	0	0
-13.5 $< T <$ 0.5	0	0	1
0.5 $< T <$ 14.5	0	1	0
14.5 $< T <$ 28.5	0	1	1
28.5 $< T <$ 42.5	1	0	0
42.5 $< T <$ 56.5	1	0	1
56.5 $< T <$ 70.5	1	1	0
70.5 $< T$	1	1	1

V. DISCUSSION

A. Temperature Sensor

Table I shows the ΔV to ΔT ratio was relatively small, with only a total change of $174mV$ from $-25°C$ to $100°C$. This meant the ADC had to be well designed if it was to give an output with high precision. It would be beneficial to linearly amplify this output voltage instead, to provide a larger voltage sweep for the ADC and thus higher precision in binary output.

However, the relation between the tension and the temperature is linear with a great correlation coefficient. This means that with a really good ADC, we could determine the temperature extremely precisely without needing a look-up table or a costly non-linear computation.

B. ADC

The Voltage ladder worked exactly as expected at high voltages. At lower voltages the output values experienced a positive skew such that the highest value (where all outputs were logic high) was greater than V_{ref}. This was unexpected but ultimately didn't affect precision of results as the resistors could be calibrated to tune the ADC to specific input voltage ranges.

As mentioned previously, the Encoder showed abnormal functioning at low-voltage, high-temperature conditions. Fixing this issue was not a priority, in the context of the design being part of an Analog Circuits course, instead the encoder was able to operated on 1.8V by altering how the ADC and temperature sensor were connected as shown in Fig. 12.

C. Complete Circuit

One can see from Table III the binary output gives an uncertainty of $\pm 7°C$ and operates over a region from $-13.5°C$ to $70.5°C$ This is a somewhat low precision but could be vastly improved by implementing a new ADC that provides more than eight possible outputs. However, the circuit may have some applications such as measuring outside temperature when only a low precision is required, in addition it can determine whether water will boil at high altitudes. For these uses, running the sensor on a lower VDD would be beneficial and so has been included in future work.

VI. FUTURE WORK

We first need to simulate the full sensor at every corner and we need to do a Monte Carlo simulation to ensure that the design would work as expected. Moreover, we haven't measured the current consumption of the sensor. Then, some power optimisations could be done.

We haven't presented any layout corresponding to our design. Then, the natural following step of our work is to create this layout. The layout should respect the design we presented and take care of layout rules, such as spacing rules and the use of dummy transistors.

The design of the temperature sensor presented in this paper was simulated using only an ideal model of each component (resistors, capacitors and transistors), without taking care of parasitic capacitance. Then, there is a need to simulate the design after having drawn the layout. This simulation would be closer to the real behaviour of a potential chip and then, can show some needs for modification.

This temperature sensor cannot be used alone on a chip: it needs other components such as a power supply. The design of this power supply can be also challenging because noise on the VDD line can lead to errors in the estimation of the temperature.

Another improvement to the design would be optimising the digital encoder. This would allow for design to be operated on lower power supply to be usable in more applications. In addition, the 'D' output could be utilised to provide a 4-bit output with higher precision or temperature range depending on the intended use.

VII. CONCLUSION

This paper details a design that uses a current proportional to temperature circuit to measure a tension difference across a resistance. The tension is passed to a flash ADC to provide a parallel binary output after being decoded.

This design of temperature sensor gives a 3-bits digital output with a range of $-13.5°C$ to $70.5°C$. The precision of the output is approximately $14°C/\text{bit}$.

Appendix A
Schematics from xschem

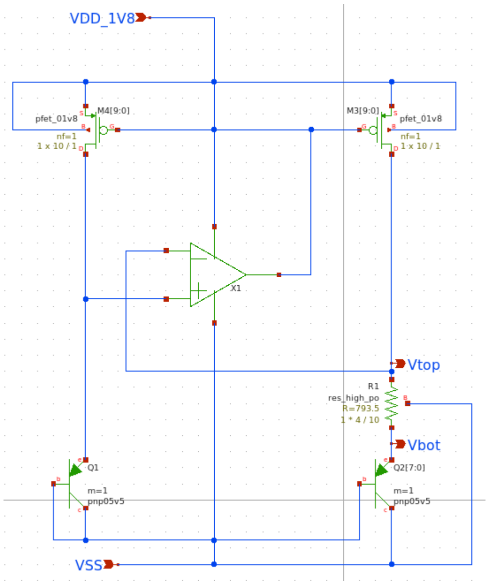

Fig. 4. Implementation of the PTAT circuit on xschem

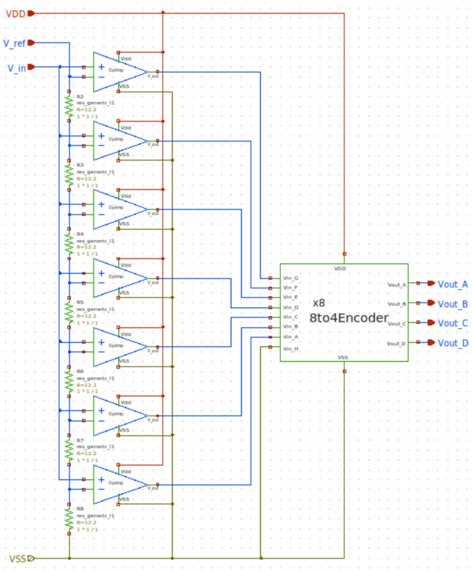

Fig. 5. Analog to Digital Converter

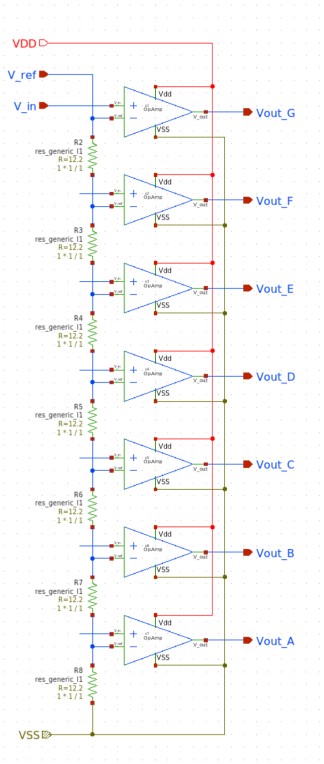

Fig. 6. Voltage Ladder.

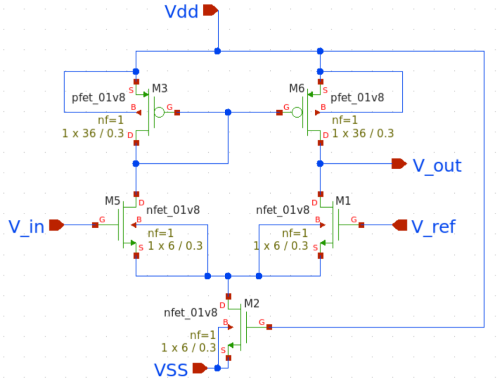

Fig. 7. OpAmp used as Comparator

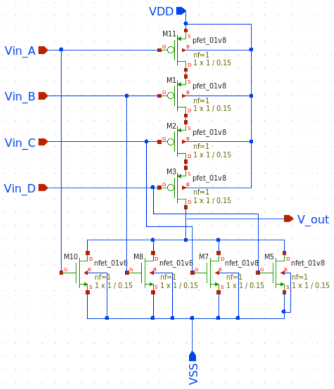

Fig. 8. CMOS OR-gate

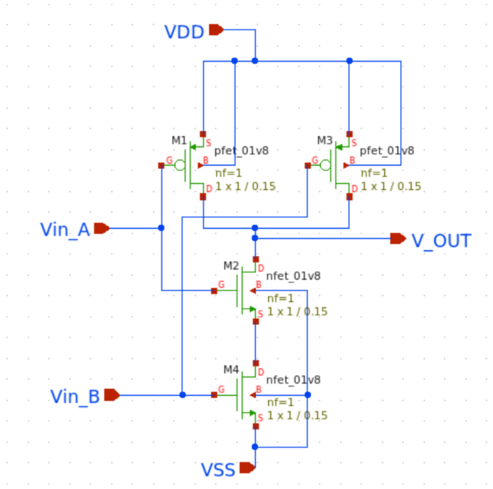

Fig. 9. CMOS NAND-gate

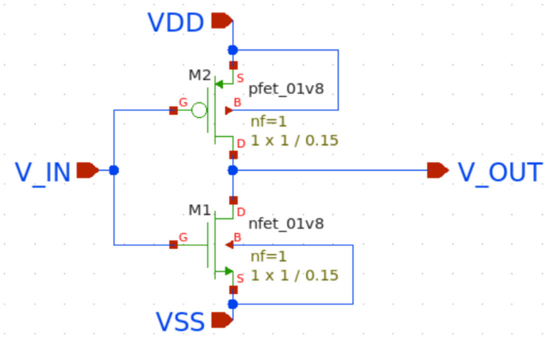

Fig. 10. CMOS Inverter

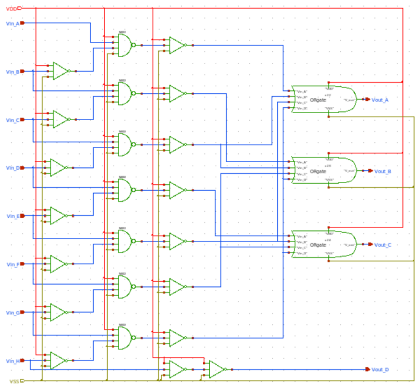

Fig. 11. Encoder Implemention

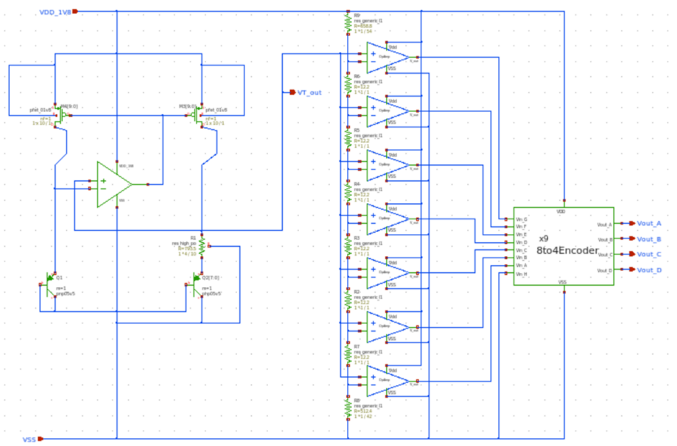

Fig. 12. Complete Temperature Sensor Design

Appendix B
Plots and Tables

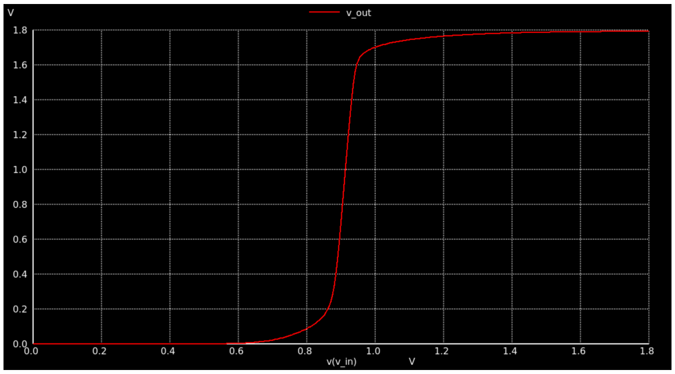

Fig. 13. V_{out} vs V_{in} of comparator implemented with OpAmp. Where $V_{ref} = 0.9V$

TABLE IV
Output tension (in mV) across the resistor for all corner at different temperature

T (°C)	Typ	Min	Typ	Max	-3std	mean	+3std
-25	48.4	42.2	50.1	54.8	37.4	48.4	59.5
0	55.4	52.8	57.3	62.7	44.8	55.9	66.9
27	61.6	58.6	64.3	70.8	49.6	61.7	73.8
50	66.9	63.5	69.9	77.5	57.4	67.9	78.3
75	73.0	69.0	76.4	85.1	63.3	73.3	83.4
100	79.6	74.6	83.6	93.2	65.3	78.9	92.5

Appendix C
GitHub Repository

The project being open source, we decided to upload most of our work publicly on GitHub at https://github.com/MathisBonnard/TFE4188_Project.

If you want to rebuild the design, you will probably need to rebuild some schematics because of path errors.

REFERENCES

[1] C. Wulff, ‘Lecture 3 - References and bias’, *aic2023*, Jan. 26, 2023. https://wulffern.github.io/aic2023/2023/01/26/Lecture-3-References-and-bias.html (accessed Apr. 28, 2023).

[2] T. C. Carusone, D. Johns, K. W. Martin, and D. Johns, *Analog integrated circuit* design, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2012.

[3] C. Wulff and T. Ytterdal, ‘A Compiled 9-bit 20-MS/s 3.5-fJ/conv.step SAR ADC in 28-nm FDSOI for Bluetooth Low Energy Receivers’, *IEEE J. Solid-State Circuits*, vol. 52, no. 7, pp. 1915–1926, 2017, doi: 10.1109/JSSC.2017.2685463.

Brandon Lawrence is a third-year student studying a double degree in Physics, and Automation and Robotics at the University of Western Australia. He is currently completing an exchange semester with the faculty of Information Technology and Electrical Engineering at NTNU.

Mathis Bonnard is a fourth-year student in the master Electronic System Design at NTNU. He is in double degree with the french engineering school Ecole centrale de Nantes where he studied Product Engineering.