# Problem Statement

This project focuses on understanding the performance tradeoffs between DLS and CMOS logic for use in a SPS. The key question for a designer deciding between DLS and CMOS logic is that given a certain power budget, is it more energy efficient to continuously run a DLS-based circuit, or to duty-cycle a static CMOS-based circuit? Additionally, what system-level overheads are induced in the static CMOS circuit (or avoided in the DLS circuit) from using the duty-cycled operation, such as large ripple on the storage node?

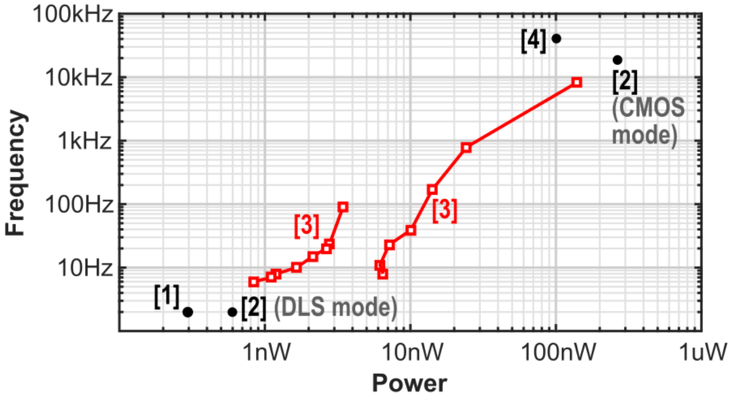

Figure 1: Comparison of DLS and CMOS (Power consumption vs frequency)

# Product Vision

In general, the power-energy tradeoff between DLS and static CMOS logic is important to consider for SPS with 1) low harvested power, 2) very compact form factor, and 3) low-frequency sampling requirements

A specific product we would like to propose is a solar-powered temperature/humidity sensor for dimly-lit spaces. This could be applied to many different scenarios, but a specific context is to monitor HVACs for leaking. The idea comes from experience, as Nojan's HVAC recently stopped working and started leaking. Nojan did not know about this until water started flooding onto the floor outside of the closet because he does not ever open the closet door that contains the HVAC. A solar-powered temperature/humidity sensor utilizing DLS logic would be perfect for this context. The system would be placed in a closet, which is dimly-lit, resulting in low harvested power from the solar cell. Although there is low harvested power, the DLS logic would ensure that the capacitor voltage would stay relatively the same once the capacitor is fully charged. When using CMOS logic, the capacitor voltage droops upon every data transmission, which could result in there not being enough power available for the next data transmission. DLS works best when used with low-frequency sampling, which would be acceptable for an application like this. The system would simply take a reading every 5-10 minutes and transmit the data to an app. This application is best done with DLS logic because it is reliable with low harvested power, as will be the case with lux available to a solar cell in a closet, due to the fact that there will be little to no droop of the capacitor voltage.

## Milestones

Based on our approach and intended results, some milestones for accomplishing the project are:

| **Week** | **Tasks** | **Deliverables** | **Status** |
|---|---|---|---|
| 10/8 - 10/15 | Define key operating scenarios (corresponding to knobs/tradeoffs) and figures of merit for assessing the research question [Nojan]<br>Design load components based on circuit models for DLS and CMOS logic that reflect leakage and dynamic power [Daniel and Nojan] | Graph of capacitor voltage with modulated operation period<br>Duty-cycled digital circuit load model with tunable parameters<br>Figure showing DLS vs. CMOS energy efficiency | Complete |
| 10/15-10/22 | Benchmark solar cells [Nojan]<br>Allow modulation of Vdd in order to view effects on capacitor voltage to observe behaviors of DLS and CMOS [Daniel and Nojan] | Benchmark plots of solar cells<br>Updated DLS vs. CMOS graph with modulating Vdd | Complete |
| 10/22-10/29 | Finalize solar cell that will be used [Nojan] | Benchmark plot for MCU<br>Updated DLS vs. CMOS graph to | Complete |

| | | | |
|---|---|---|---|
| | Benchmark DLS and CMOS MCU's for load [Nojan]<br>Add functionality to comparison framework to sweep other parameters between operations [Daniel and Nojan] | observe parameter sweeps | |
| 10/29-11/05 | Benchmark components (Moisture sensor and Bluetooth TX) that make up the load of the product [Nojan]<br>Sweep energy vs repeat-rate (twhole) to see break points for DLS and CMOS and setting up more design sims [Daniel and Nojan] | Benchmark plots for bluetooth tx and moisture sensor<br>Energy vs. repeat-rate plot that highlights the breakpoint for DLS vs. CMOS | Complete |
| 11/05-11/12 | Model SPS with decided load system using DLS logic for MCU [Daniel and Nojan] | Figures showing performance of proposed SPS | Complete |
| 11/12-11/17 | Sensitivity analysis of FoM to the different operating scenarios and knobs related to the digital circuits [Nojan]<br>Analyze the results and summarize the key takeaways for SPS design [Daniel] | Defined key takeaways for SPS design when considering DLS vs. CMOS | Complete |
| 11/18 - 11/26 | Sweep VDD and twhole to see, for each VDD, what twhole the system would have to be | Breakeven plot with multiple parameters considered | Complete |

| | | | |
|---|---|---|---|
| | operating at to consider using a DLS as a “win” [Daniel and Nojan]<br>Make final presentation slides [Daniel and Nojan] | Final slides | |
| 11/27-end | Work on final paper and finalizing progress tracker | Complete paper<br>Complete progress tracker | Complete |

## Modeling

We want to observe the following trends with the model:

1. Voltage at the energy storage capacitor over time in order to investigate the transient effects of the digital circuit duty-cycling
2. the power and energy consumed of both the digital circuit and the full SPS while the data sample is being processed and over a longer period of time

In order to fully characterize the tradeoffs between DLS and CMOS, we must extend the model in the following ways:

1. Add the capability to build a duty-cycled digital circuit load model with tunable parameters for simulation
    a. This has been completed
2. Allow transient simulation of energy use comparing DLS and CMOS digital circuit
    a. This has been completed
3. Allow modulation of the operating period of load in order to view effects on capacitor voltage
    a. This has been completed
4. Calculate active power of circuit based on Vdd, operating period, and capacitance
    a. This has been completed
5. Allow modulation of Vdd, alongside other modulating characteristics, in order to view effects on capacitor voltage

In order to show the feasibility of our proposed product, we must utilize the model in the following ways:

1. Model power consumption of the load system we plan on using with always-on DLS
2. Model harvested power based on measurements of lux available in dimly lit spaces
3. Modulate parameters in order to see the performance of our product

*Our python code:*
Our code can be found here. In order to access the code, one of us will have to grant you access as it is a private repository. We have built the following files in order to extend the model:

- PythonModeling.py
  - Can be ignored, we were just getting started with the model
- PythonModeling2.py
  - Producing full load waveform using hardcoded power values
- PythonModeling3.py
  - Modulating operation period and looking at capacitor voltage
- PythonModeling4.py
  - Comparing power and energy efficiency of DLS and CMOS circuits
- PythonModeling5.py
  - Can be ignored, setting up framework for design space sims
- PythonModeling6.py
  - Integrating Daniel's leakage model for static-CMOS and DLS
  - Daniel's leakage model not included in repo
- PythonModeling7.py
  - Showing total energy consumption per time interval versus circuit supply voltage
  - Showing total energy consumption per time interval versus wait time interval Twait
- PythonModeling8.py
  - Getting line crossings for energy consumption graphs to construct breakeven points for graph
- Final Project Modeling.ipynb
  - Jupyter notebook used to model system for proposed application and compare DLS and static-CMOS MCUs

## Key Results

As can be seen in Figure 2, we were able to modulate the same load with different operating periods in order to see the effect that the operating period has on the voltage on the capacitor.

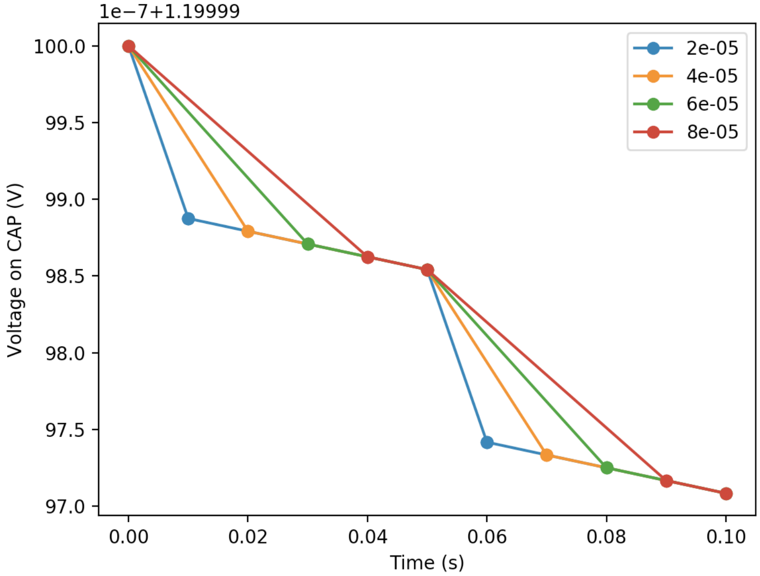

Figure 2: Capacitor voltage with modulating operation period (t_clk)

Figure 3 shows a duty-cycled digital circuit load mode. The model assumes that a processor activates and runs Nop number of clock cycles at a rate tclk, and then enters standby (clock gating but not power gating) for a waiting period Twait. All of these parameters are tunable in simulation. The active power is calculated based on the chosen t_clk value as well as the effectice switched capacitance value and supply voltage Vdd. The load profile in Fig. 3 shows a Nop of 500, a tclk of 500us, and a Twait of 5 minutes.

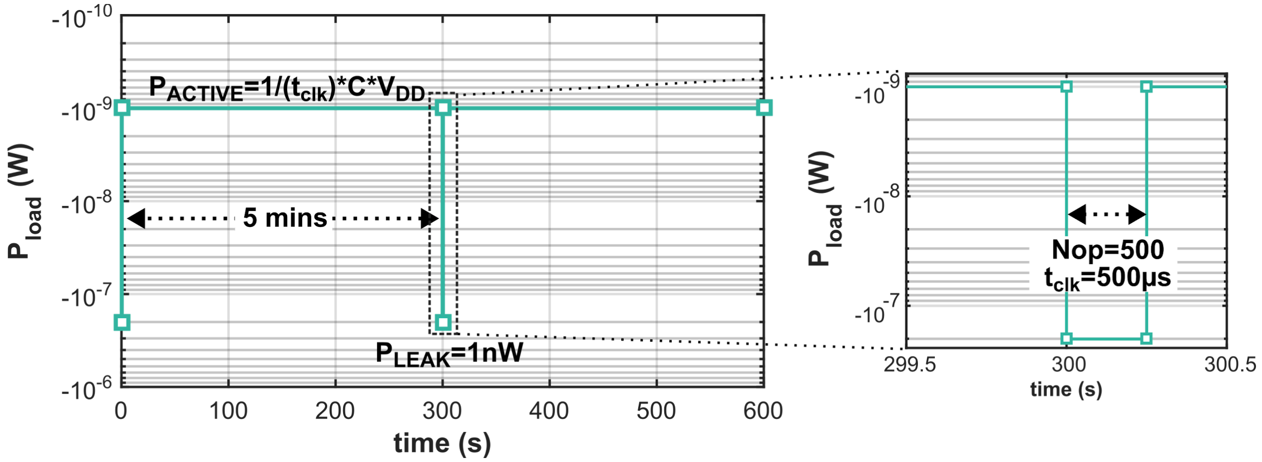

Figure 3: Duty-cycled digital circuit load model with tunable parameters

Fig. 4 shows the capacitor voltage versus time using the load profile model that was shown above. In the plots below, the DLS circuit is running with a 2kHz clock and the CMOS circuit is running with a 50kHz clock. Both circuits have a supply voltage 0f 0.5V, a Nop workload of 500, and a Twait of 5 minutes. Under these conditions, the CMOS finishes its workload in 10ms using 256nJ of energy, while the DLS processor takes 250ms and 1uJ to complete the workload. However, once each circuit is finished with its own workload, it enters standby until the 5 minute wait time is up. The DLS has less leakage than the CMOS, so by the end of the full interval, the DLS is actually more energy efficient.

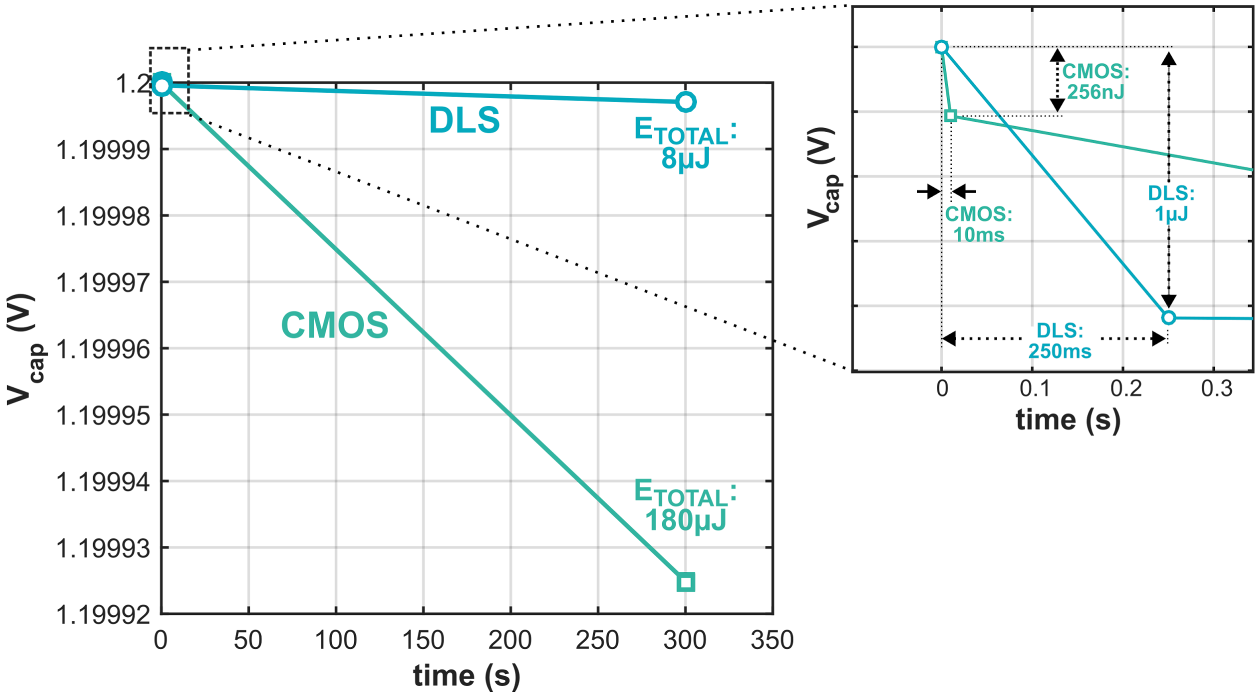

Figure 4: DLS vs. CMOS energy efficiency

In the previous work, we used a model to calculate the dynamic power based on the circuit parameters (Vdd, tclk) but we were using an arbitrary constant value for leakage power. Since the last report, we have added a leakage model for both CMOS and DLS logic that calculates the leakage power based on the chosen supply voltage, which allows for our load power model to be more realistic.

The results from Fig. 4 show the total energy consumed by the DLS and CMOS circuit over the full time interval. This energy includes the dynamic energy that is initially consumed by the processor as well as the leakage energy that accumulates during the remainder of the wait interval. We analyze how this total energy consumption changes as a function of the effects of the load profile parameters (Nop, tclk, Twait) and circuit parameters (Vdd, Csw).

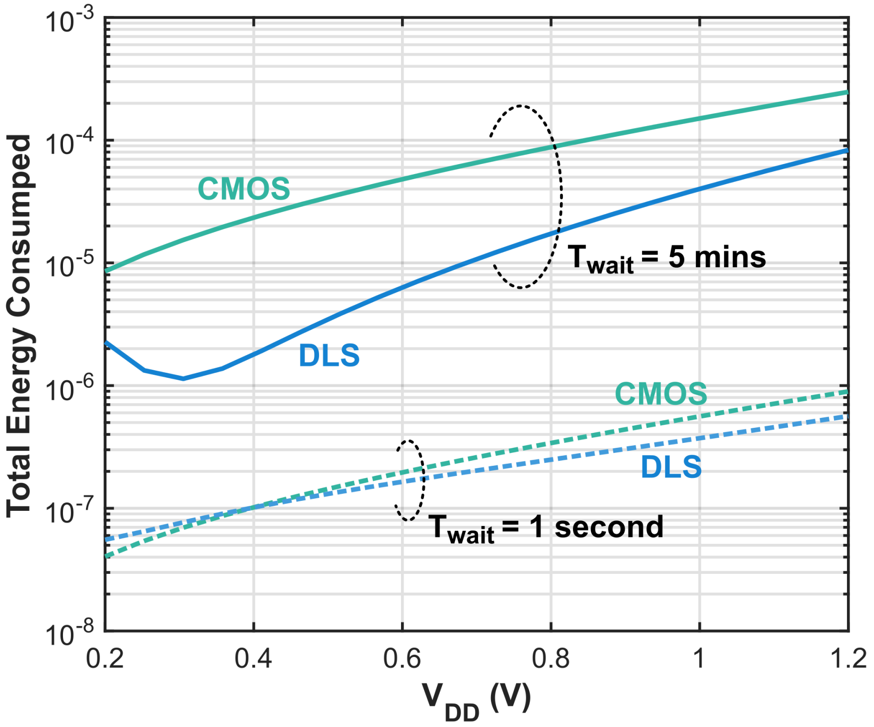

Fig. 5 - Total energy consumption per time interval versus circuit supply voltage. This is shown for two different Twait intervals of 1 second and 5 mins. All other parameters are held constant (Nop=500, fclk=50kHz for CMOS and 2kHz for DLS).

Fig. 5 shows the total energy consumption versus circuit supply voltage for two different Twait values. When Twait=5 minutes, the circuits spend large amounts of time in standby after they complete their operations, so leakage energy dominates the total energy usage (as shown in Fig. 4 above), so DLS is more energy efficient due to its lower leakage. When Twait is decreased to 1 second, the leakage reduction benefit from DLS is negated so it does not reduce energy much beyond the CMOS. For low Vdd where the CMOS is most energy efficient, it actually outperforms the DLS logic.

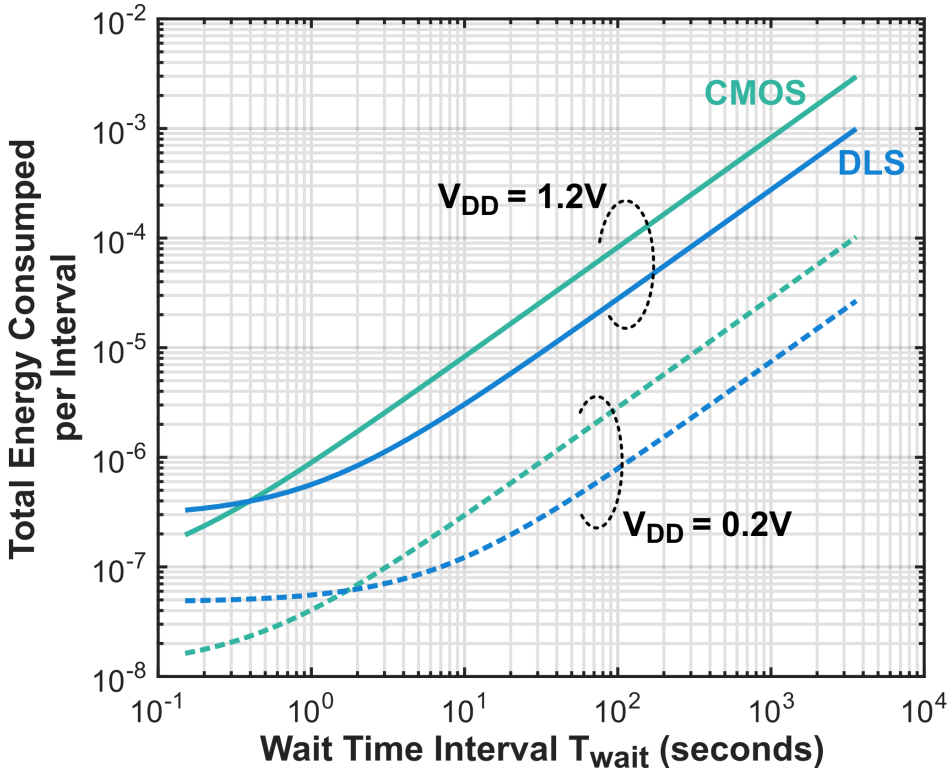

Fig. 6 - Total energy consumption per time interval versus wait time interval Twait. This is shown for two different Vdds of 0.2V and 1.2V.

As Fig. 5 shows, there is a certain Twait for a given Vdd that causes the CMOS to be more energy efficient than DLS. In particular, Fig. 5 shows that at 0.4V Vdd, a Twait of 1s or greater is required for DLS to be more energy efficient. Fig. 6 shows an alternate view of this by sweeping Twait for two different Vdd values of 0.2V and 1.2V. For each Vdd, a different Twait is required for the DLS to surpass the CMOS energy efficiency. Increasing Twait significantly past the critical point does not continue to reduce DLS energy relative to the CMOS logic since their energies both become dominated by leakage power.

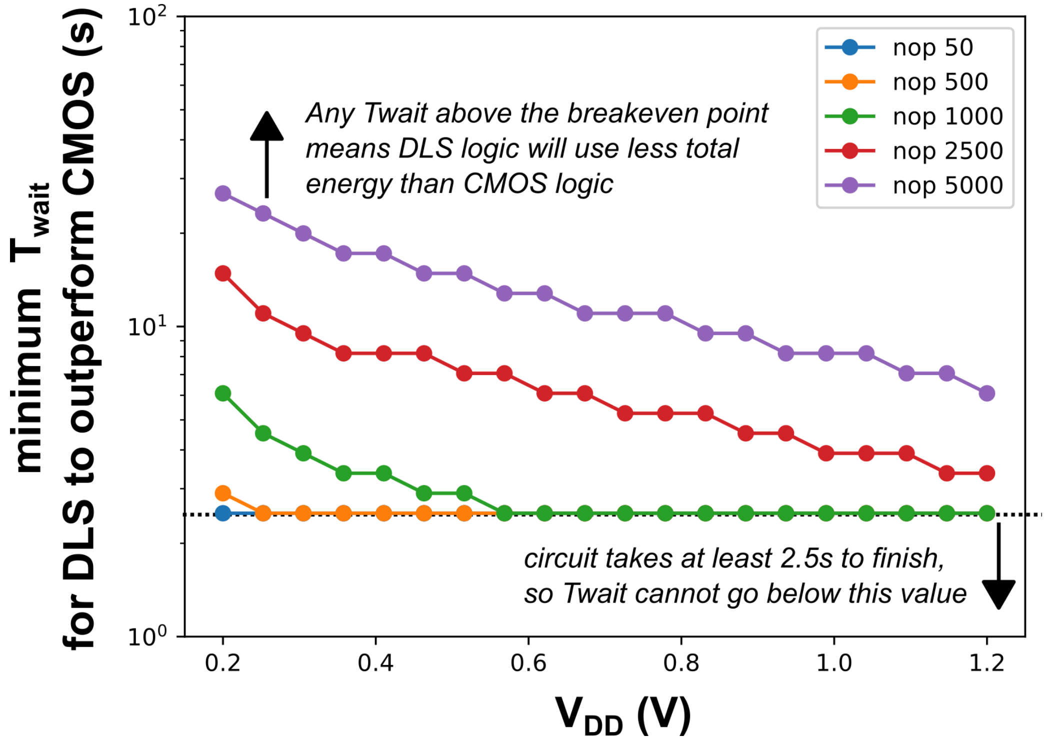

Fig. 7 - Breakeven points for varying supply voltages, number of operations, and wait time intervals

Fig. 7 shows a conclusive way for circuit designers to know when DLS is a “win”, “not win”, or will fail when compared to static-CMOS. As the graph shows, for the multiple lines on the graph representing the number of operations, any wait time interval above the points on a certain line yields a “win” for DLS. Any wait time interval below the points on a certain line yields a “not win” for DLS. There is a lower limit shown by the dotted line that shows the failure point for DLS. DLS will not successfully operate at any wait time interval below the dotted line.

**Tradeoffs**

Three key constraints that can be addressed using DLS logic are low harvested power, form factor, low-frequency sampling. Utilizing DLS in a project like this introduces a lot of tradeoffs:

1. There is no need for duty cycling when using DLS because DLS logic allows you to get low power without duty cycling. As was mentioned in class, everything comes at a cost, so not duty cycling would allow for less power consumption, but would result in an always-on circuit. To model this tradeoff, we could measure the power consumption of duty cycling with a CMOS-based circuit and compare it to the power consumption of keeping a DLS-based circuit always on.
2. A smaller capacitor would be used, but there would be less droop on the capacitor voltage. The tradeoff here would be capacitor size vs. capacitor voltage droop. This can be modeled by comparing the capacitor size to the droop on the capacitor voltage on both DLS-based and CMOS-based circuits.

3. DLS-based circuits are most reliable when working at a low sampling frequency, which is why the application does not have a high sampling frequency. The tradeoff here is sampling frequency vs. system reliability. We chose to have a low sampling frequency in order to have high system reliability. In order to model this, we can measure the relationship between the number of operations that have enough power to transmit data vs. the sampling frequency for both CMOS and DLS-based circuits.

## Final Paper

Our final paper can be found here.

# 4501 Material

## Block Diagram

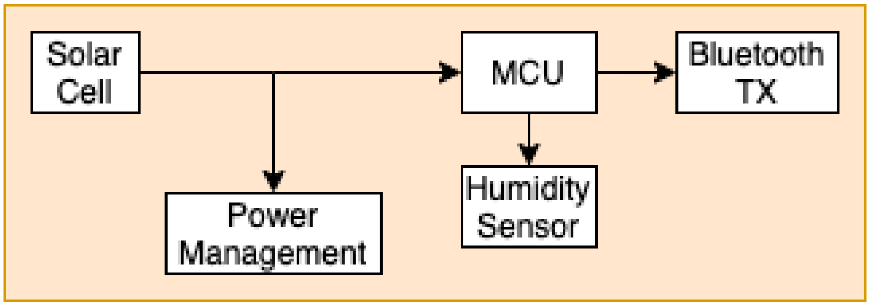

Fig. 8 - Block diagram of proposed product

This system utilizes a solar cell to power the system. We “black box” the power management unit for this system in the design process. The MCU receives data from the humidity sensor and transmits that data using a bluetooth transmitter.

## Harvesting Environment

We could not find online datasets that contain lux or irradiance measurements for our ideal setting, which would be a closed closet. While we couldn’t find data for this setting, there were two datasets that had gathered irradiance and lux data for multiple different spots in an office setting: Intel Lab Data and Columbia Indoor Light Energy. The Intel lab data did not have any spots that were getting noticeably little light, but the Columbia indoor light energy source had a sensing node placed on a bookshelf, so it would be getting very little light. They presented irradiance graphs for this:

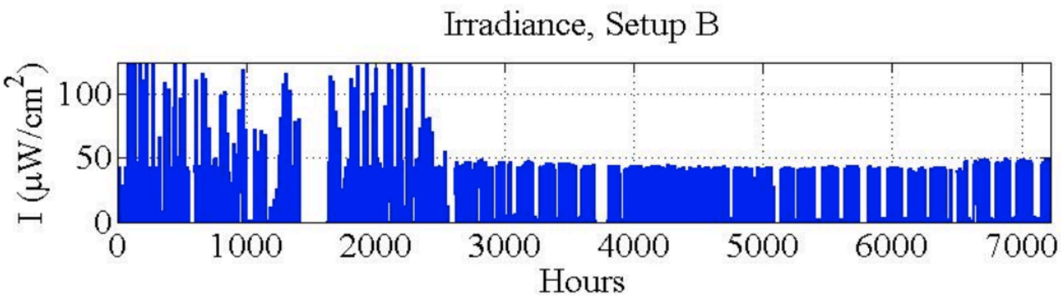

Fig. 9 - Irradiance measurement from bookshelf

Fig. 9 shows that the average irradiance that a bookshelf in an office building is exposed to ranges from ~50 uW/cm^2 to ~125 uW/cm^2. Relative to the amount of power a solar cell can harvest outdoors or even indoors, this is pretty low. Also taking into account that we will be using a solar cell that is relatively low size and in a setting that is most likely darker than a bookshelf in an office building, we mainly use this data to show that we will be working with even less irradiance than is shown in figure 9. Given more time and proper resources, we would have been able to actually gather traces from a real closet over a week to show the trends of harvestable power available to a solar cell.

# Benchmarking

### Microcontrollers

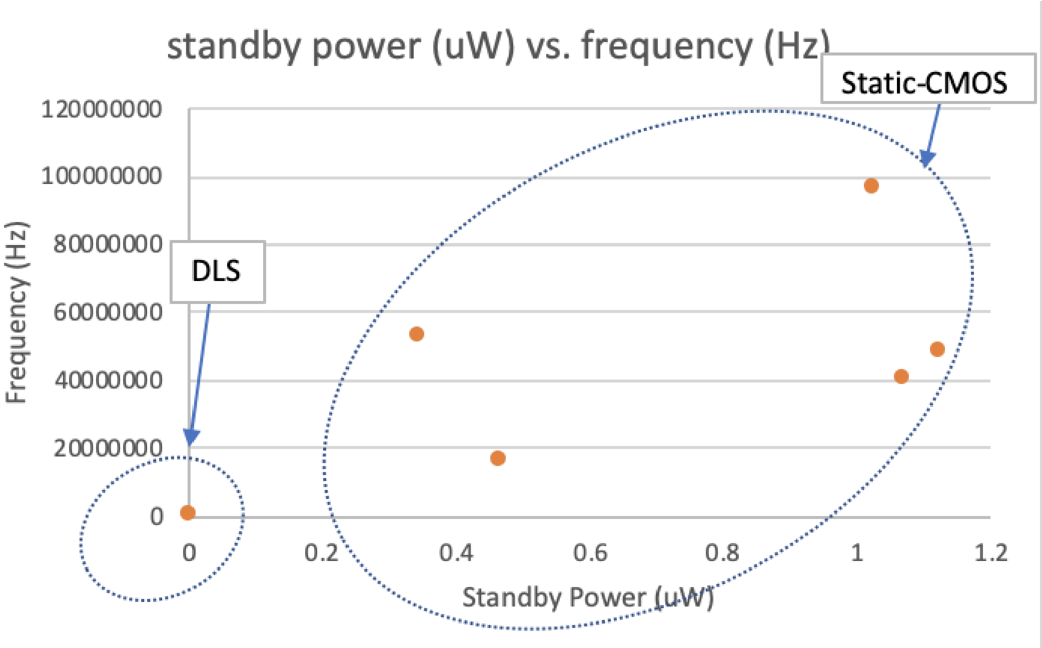

Fig. 10 - MCU Benchmarking

This plot shows the tradeoff between frequency and standby power. DLS is known to suppress standby power, as can be seen by the lower left point. Due to the nature of our application, DLS is preferable to minimize standby power. Although this lowers the frequency, it still meets the requirements for our system.

### Solar Cells

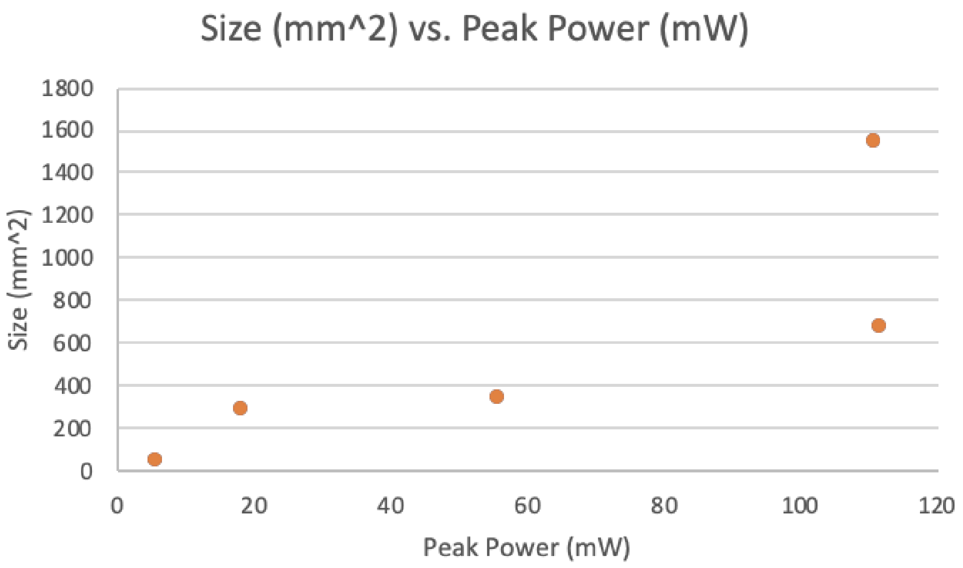

Fig. 11 - Solar Cell Benchmarking

One of the benefits of using DLS logic is the smaller form factor that can be utilized. Once we initially charge our energy storage, we will not notice much droop upon each operation period. Because of the small form factor, we want to use a smaller on-chip solar cell that still has a relatively high power output.

### Humidity Sensor

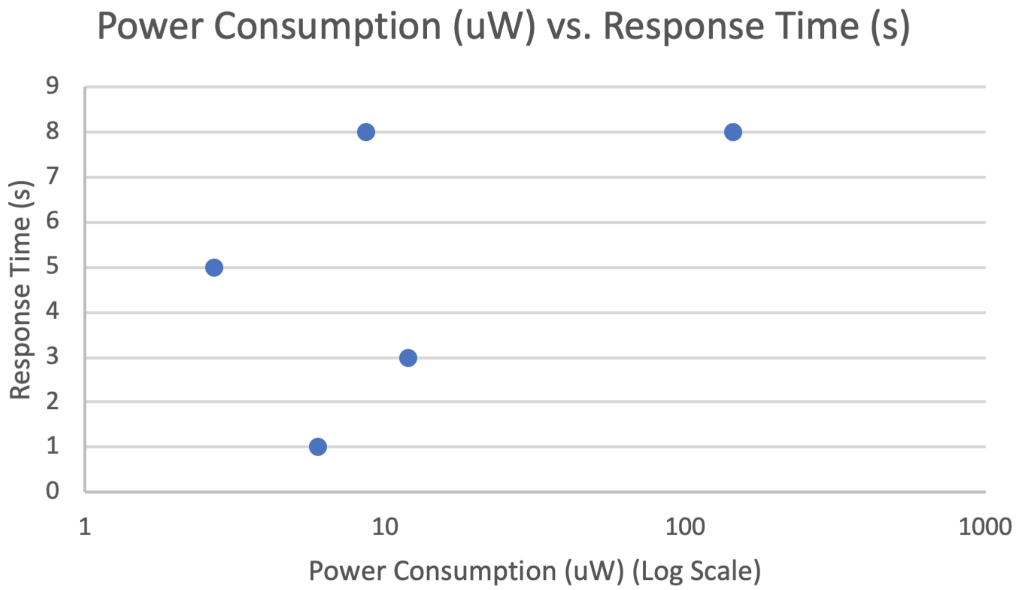

Fig. 12 - Humidity sensor benchmarking

For the humidity sensor, one of the main tradeoffs is power consumption and response time. We want to get a near-instant response time while not consuming too much power.

**Bluetooth Transmitter**

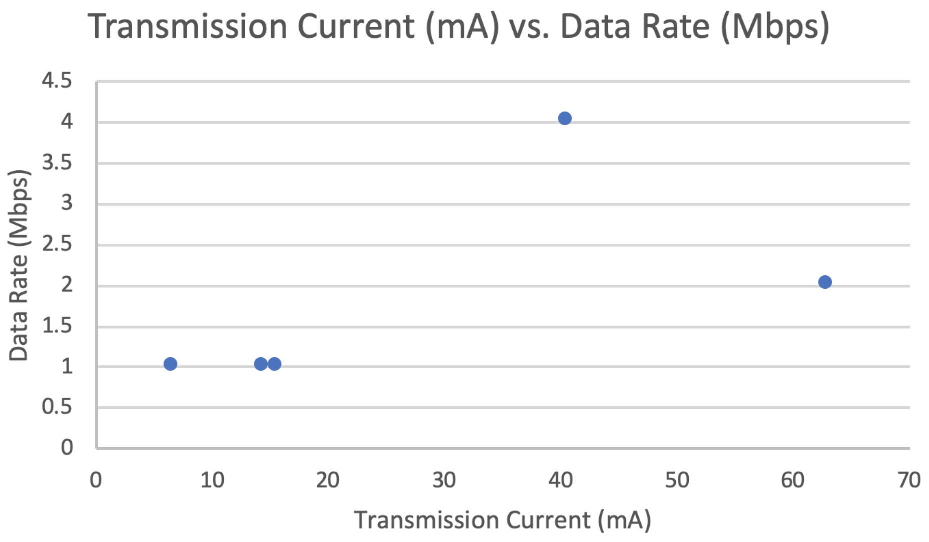

Fig. 13 - Bluetooth transmitter benchmarking

For the bluetooth transmitter, we don't need a high data rate, so power consumption, which is directly related to transmission current, is important.

## Selected Components

| Component | Link | Justification |
|---|---|---|
| Solar Cell | http://ixapps.ixys.com/DataSheet/KXOB22-12X1F_Nov16.pdf | Relatively high peak power with relatively low size, 22% efficiency |
| MCU | https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8822384 | DLS processor, low frequency, but has very low leakage |
| Bluetooth Transmitter | https://www.nordicsemi.com/-/media/DocLib/Other/Product_Spec/nRF8001PSv13.pdf | We don't need a high data rate, we just need very low power and this fits that spec |
| Humidity Sensor | https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280-DS002.pdf | Low power consumption and 1 second response time |

## Final Product Simulations

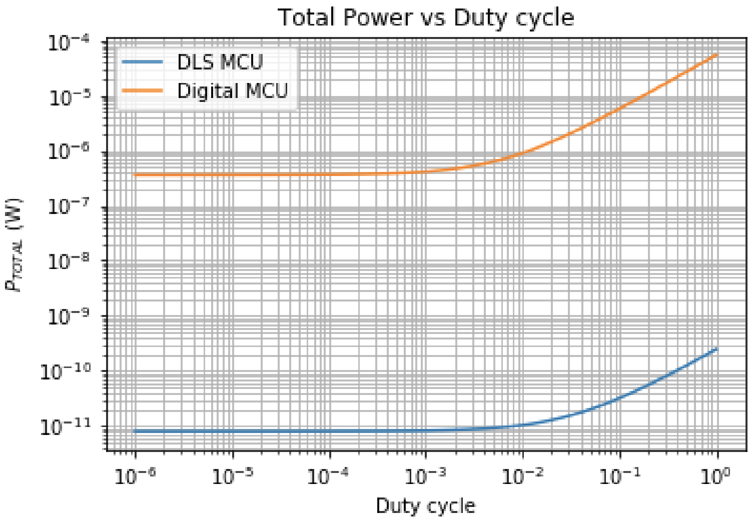

Fig. 14 - Comparison of leakages of DLS and static-CMOS MCUs

As can be seen from the figure above, the DLS MCU that we will utilize for this product offers a 100,000x reduction in leakage power.

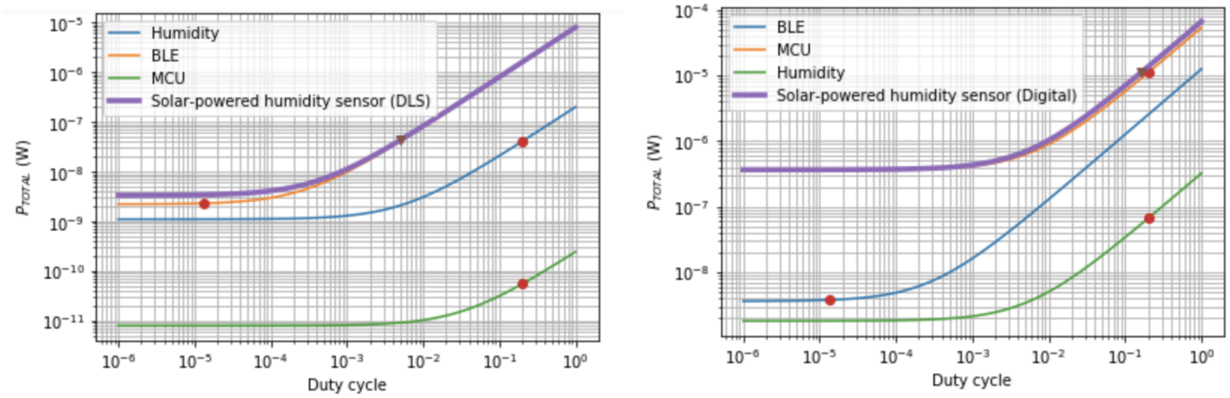

Fig. 15 - Comparison of proposed product with DLS MCU and static-CMOS MCU implementations

The figures above show the power consumption of the solar-powered humidity sensor when using a DLS MCU and a digital MCU. Using the DLS MCU results in a 100x system-wide reduction of leakage. When looking at the digital-based system, the leakage mainly consists of the leakage that is associated with the digital MCU. When looking at the DLS-based system, the leakage is mainly driven by the Bluetooth transmitter.

## Datasheet

The datasheet can be found here.

## Conclusions about the proposed product

This product should be implemented using DLS due to the low leakage that DLS offers to a system. This system will have a relatively long Twhole, which we have shown warrants the use of DLS in a system. More modeling would have been done with more time or group mates, but Daniel and I made the decision that the work we were doing for 6501 was more valuable for advancing knowledge in the field. For the application side, we thought it would still be valuable to show the leakage reduction that results from using DLS on a MCU and system level.

## References

**Lit Review**

[1] W. Lim, I. Lee, D. Sylvester, and D. Blaauw, “Batteryless sub-nW cortex-M0+ processor with dynamic leakage-suppression logic,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, San Francisco, CA, USA, 2015, pp. 1–3.

[2] L. Lin, S. Jain, and M. Alioto, “A 595pW 14pJ/cycle microcontroller with dual-mode standard cells and self-startup for battery-indifferent distributed sensing,” in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), San Francisco, CA, USA, 2018, pp. 44–46.

[3] D. S. Truesdell, J. Breiholz, S. Kamineni, N. Liu, A. Magyar and B. H. Calhoun, "A 6–140-nW 11 Hz–8.2-kHz DVFS RISC-V Microprocessor Using Scalable Dynamic Leakage-Suppression Logic," in *IEEE Solid-State Circuits Letters*, vol. 2, no. 8, pp. 57-60, Aug. 2019.

[4] S. Hanson et al., “A low-voltage processor for sensing applications with picowatt standby mode,” IEEE J. Solid-State Circuits, vol. 44, no. 4, pp. 1145–1155, Apr. 2009.

[5] B. H. Calhoun, S. Khanna, Y. Zhang, J. Ryan and B. Otis, "System design principles combining sub-threshold circuit and architectures with energy scavenging mechanisms," *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, Paris, 2010, pp. 269-272.

[6] B. H. Calhoun, A. Wang and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1778-1786, Sept. 2005.

**MCU Components**

- http://www.ti.com/lit/ds/symlink/msp432p401r.pdf
- https://www.analog.com/media/en/technical-documentation/data-sheets/ADuCM4050.pdf
- https://datasheets.maximintegrated.com/en/ds/MAX32660.pdf
- http://ww1.microchip.com/downloads/en/DeviceDoc/40001607D.pdf
- https://www.silabs.com/documents/public/data-sheets/efm32pg1-datasheet.pdf
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8822384

**Solar Cells**

- http://ixapps.ixys.com/DataSheet/XOD17-Solar-Cell-Die-Datasheet_Mar-2008.pdf
- http://ixapps.ixys.com/DataSheet/KXOB22-12X1F_Nov16.pdf
- http://ixapps.ixys.com/DataSheet/SLMD600H10L_Nov16.pdf

**Moisture Sensors**

- https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/0_Datasheets/Humidity/Sensirion_Humidity_Sensors_SHTW2_Datasheet.pdf
- https://ams.com/documents/20143/36005/ENS210_DS000470_2-00.pdf
- http://www.ti.com/lit/ds/symlink/hdc2010.pdf
- https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=HPC199_6&DocType=Data+Sheet&DocLang=English
- https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280-DS002.pdf

**Radios**

- https://media.digikey.com/pdf/Data%20Sheets/Texas%20Instruments%20PDFs/CC256x_Rev_B.pdf
- http://toshiba.semicon-storage.com/content/dam/toshiba-ss/emea/en_gb/bluetooth/data/TC35661SBG-501_E_rev100_Oct_2013_Overview_Specification.pdf
- https://www.nordicsemi.com/-/media/DocLib/Other/Product_Spec/nRF8001PSv13.pdf
- https://www.nordicsemi.com/-/media/DocLib/Other/Product_Spec/nRF8002PSv11.pdf
- https://media.digikey.com/pdf/Data%20Sheets/Cypress%20PDFs/CYBL1xx7x_RevK_3-27-17.pdf