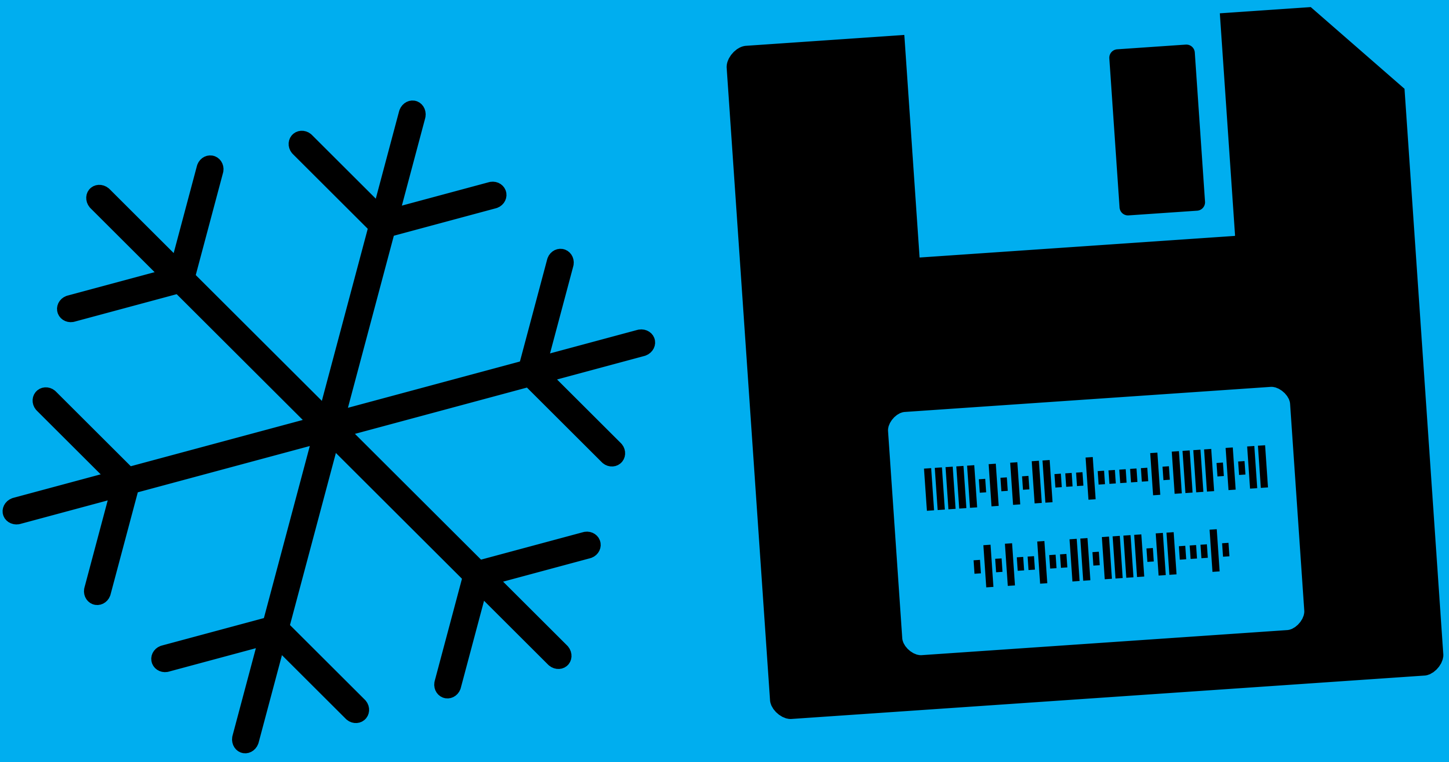
Cryogenic Digital CMOS Memories for Quantum Computing

R.A. Damsteegt

Technische Universiteit Delft



TU Delft

Cryogenic Digital CMOS Memories for Quantum Computing

by

R.A. Damsteegt

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 29, 2022 at 10:00 AM.

Student number:	4573625	
Project duration:	September 1, 2021 – August 29, 2022	
Thesis committee:	Dr. F. Sebastiano,	TU Delft, supervisor
	Dr. ing. R. K. Bishnoi,	TU Delft
	Prof. dr. ir. G. Gaydadjiev,	TU Delft

This thesis is confidential and cannot be made public until August 29, 2027.

Abstract

Scalable universal quantum computers require classical control hardware, physically close to the quantum devices at cryogenic temperatures. Such classical controllers need digital memory for various applications, ranging from high-speed queues to high-speed and low-speed lookup tables and working memory. The power consumption of the memories should be within the available cooling power at these temperatures. To obtain the best memory design with the lowest power consumption, cryogenic-CMOS characteristics need to be taken into account during design. This thesis aims to develop a model that can be used to find the optimal memory cell design for each application, while taking area, latency, error rate, and power constraints into account. A model is developed to estimate the error rate and power consumption of a memory core for four cell designs, namely three embedded dynamic cell designs and one static cell design, over a range of applications in terms of memory operation frequency and read/write operation ratio. The model is constructed using room temperature and $233\,\mathrm{K}$ simulation data of individual cells and peripherals from a TSMC $40\,\mathrm{nm}$ technology. To estimate the error rate and power consumption at $4.2\,\mathrm{K}$, the model is extended with empirical cryogenic-CMOS characteristics, such as an increase in threshold voltage and a steeper subthreshold slope, since good device models at this temperature are not available. To verify and improve the memory model, a test chip in TSMC $40\,\mathrm{nm}$ technology is designed, which includes eight fully-custom memories with two threshold-variations of each of the four cell designs to mitigate the cryogenic-CMOS threshold voltage increase. These memories are connected to an on-chip programmable microcontroller through a bus which enables flexible and high-speed testing without the need for high-speed I/O. This chip design is taped out and will be measured to verify and refine the model. At room temperature, the static cell design outperforms the dynamic cells in all memory applications with less than 10^7 operations per second. However, at cryogenic temperatures, the embedded dynamic cell designs become feasible for applications with more than 300 operations per second, due to the significantly reduced refresh rate. The best embedded dynamic cell design depends on the read/write memory operation ratio required by the application. After verification and improvement of the memory model based on the measurement results from the test chip, this model can be used to find the best cell design for a given application based on its operation frequency and read/write operation ratio. Apart from comparing the performance of a single memory design at room temperature and $4.2\,\mathrm{K}$, this work also allows for direct comparison between the different memory cells designs, designed with the same technology, architecture, peripherals, and by the same design rules. Since only a single architecture is used with a single design for each of the peripherals, further optimisation is required for a cryogenic-optimised full memory design.

Acknowledgements

I would like to start by thanking my supervisor Dr. Fabio Sebastiano for this interesting project, which allowed me to combine my knowledge in both Electrical Engineering and Computer Engineering. I have thoroughly enjoyed the collaboration during the past year and learned a lot during the frequent meetings, both online and in person, mainly about the analog side of electronics.

Secondly, I would like to thank Ramon Overwater for being my daily supervisor. His help made the entire process a lot smoother by providing both technical support on practicalities and feedback and critical questions, either during or outside the meetings. Our shared interest in digital electronics resulted in helpful discussions on digital design and implementation.

I would also like to thank Rajendra Bishnoi and Georgi Gaydadjiev for joining my thesis committee.

I would like to thank the Coolgroup members for helpful and fun discussions whenever I went to Delft.

Finally, I would like to thank my parents, brother, sister, and friends for the support, a stable environment to work from home, and fun distractions whenever I was not working.

R.A. Damsteegt
Nieuw-Lekkerland, August 2022

Contents

1 Introduction

In this chapter, the motivation for this thesis work is explained. This is followed by the objective of this thesis. Finally, the outline of the thesis is presented.

1.1. Motivation

Quantum computers can theoretically outperform classical computers for certain problems by using superposition, entanglement, and interference of quantum states [1], [2]. The quantum states are encoded in qubits, which are implemented using quantum-mechanical systems such as, for example, single electron spins [3], [4] or superconducting transmons [5], [6]. Operations on these quantum states are executed by classical electronics that generate varying electric and magnetic fields through voltage and current pulses. The shapes, timing, and magnitudes of the pulses are determined by a classical controller running a quantum algorithm.

The classical controller that schedules and executes the operations on the quantum states requires digital memory to store programs, readout results, and waveforms of the pulses that are required to execute the quantum operations. Additionally, memory is required to hold quantum operations to align the timing of the operations, keep track of classical data and control flow, and interface with a host computer in a heterogeneous computing system. In order to build a scalable quantum computer, the classical electronics should be integrated close to the qubits [7], [8]. Since most qubit implementations currently still need to be cooled to millikelvin (mK) temperatures to limit the amount of thermal noise [9], [10], the current operational temperature of the classical electronics is proposed to be around $4.2\,\mathrm{K}$. This increases the available cooling power in fridges and allows for testing using liquid helium. This requires the digital memories to also operate at these cryogenic temperatures. Eventually, the classical electronics and the qubits are envisioned to be integrated together at an intermediate temperature.

At $4.2\,\mathrm{K}$, the available cooling power is still limited. This requires the memory power consumption to be minimised while meeting all other application constraints such as the maximum area, latency, and error rate. Due to a lack of device models that are valid at $4.2\,\mathrm{K}$, a memory model can be used to predict performance based on room temperature simulation results combined with empirical cryogenic-CMOS characteristics.

At cryogenic temperatures, transistor leakage decreases significantly [11]. Dynamic memories may provide high-density and low-power storage opportunities that can not be matched by static memories, due to the increased retention time of charge on the storage capacitance [12]–[15].

1.2. Thesis objective

In this work, a model is developed to investigate the trade-offs associated with the selection of a memory cell design at both room temperature and cryogenic temperatures. The target application of this work is various memory applications in a classical controller for quantum computing, but the results could also be useful for memory designs in spaceflight, cryogenic experiments, and high performance computing. Since memory design is an extremely broad topic, this thesis is limited to the selection of the memory cell type by comparing four cell designs: three embedded dynamic cell designs, where data is stored in the form of charge on a capacitance, and one static cell design, where data is stored in a bi-stable circuit.

Previous work on cryogenic memories mainly focused on the characterisation and comparison of single memory designs at various temperatures, from room temperature down to $4.2\,\mathrm{K}$. Due to variations in memory designs and the used technologies, results of various works can not be quantitatively compared with each other. This work aims to compare various cell designs directly by keeping the technology and architecture as similar as possible with the goal of finding the best memory cell design for cryogenic memory applications.

The model, based on simulations of individual memory components in a TSMC $40\,\mathrm{nm}$ process at room temperature and $233\,\mathrm{K}$, is developed for selection of the best memory cell design for a given application. The 'best' design is based on a combination of area, latency, error rate, and power consumption constraints. Due to the limited cooling power available at these low temperatures, a low power consumption will be the main target. This model helps to find the best cell design without having to simulate entire memory designs and includes cryogenic-CMOS characteristics without the need for complete device models.

To verify the model, a test chip is designed in TSMC $40\,\mathrm{nm}$ with eight fully-custom memory designs using the various cell designs. Two variations of every cell design are implemented using different threshold voltage devices to mitigate the increase of threshold voltage at cryogenic temperatures. The memories can be measured at both room temperature and $4.2\,\mathrm{K}$ using an on-chip programmable microcontroller, and the results can be used to verify and refine the model.

1.3. Outline

This thesis will start with necessary background on digital CMOS memories and cell designs, quantum computing and the associated digital memory applications, and cryogenic CMOS and cryogenic digital memories in chapter 2. This is followed by a modelling effort to investigate the trade-off between different memory designs at cryogenic temperatures in chapter 3. In chapter 4, the design of the actual memories for tape-out are explained. Chapter 5 covers the design of the testing architecture up to the complete top level IC design and describes the test plan. Finally, the thesis is concluded in chapter 6, together with an overview of suggested future work.

2 Background

In this thesis, the trade-off between different digital CMOS memory cell designs for applications in quantum computing is investigated. This chapter will cover the necessary background information required to understand the following chapters. First, an overview of digital CMOS memories is given, starting with a general overview of how such a memory works, followed by a more detailed description of the operation of several memory cell designs. Next, the quantum computing application will be covered with the necessity for digital memories and some of the memory requirements. Finally, cryogenic CMOS is covered with the changes to the technology at low temperatures and an overview of the literature on cryogenic memories.

2.1. Digital CMOS memories

Before going into the memory applications and the effects of cryogenic temperatures on memories, the general architecture of digital CMOS memories and the cell designs of interest are shown. The purpose of a memory is to store digital information using a write operation and being able to retrieve it after an arbitrary amount of time using a read operation. Of course, it is important that the data does not change between the write and read operation.

The structure of this section is as follows. First, the components of a general memory architecture are shown and it is explained how they operate together to form a functioning memory. This is followed by an overview of five memory cell designs that will be used throughout the remainder of this thesis with specifics about their operation.

2.1.1. Memory architecture

A basic memory architecture is shown in fig. 2.1. A two-dimensional array of cells is shown which is connected to several peripherals. Horizontally neighbouring cells (a *row* or *word*) share a *wordline* (WL), while vertically neighbouring cells (a *column*) share a *bitline* (BL). The wordlines can be seen as the control path, used to select a row of interest based on the address which is decoded by the row decoder. The bitlines can be seen as the data path, used to move data from the data input to the cells using the *bitline drivers* or from the cells to the data output using the *sense amplifiers*.

The memory has three modes of operation: *hold*, *write*, and *read*. In the hold mode, no row is selected and all cells hold the data in their internal state. In the write mode, a single wordline is driven by the row decoder and all bitlines are driven by the bitline drivers based on the input data. The cells in the selected row copy this data to their internal state, overwriting the previous state. In the read mode, again a single wordline is driven by the row decoder. However, now the selected cells develop a voltage on the bitlines based on their internal state. The sense amplifiers use the bitline voltage to determine the internal state of the selected cells, by comparing it with another bitline or a reference voltage, and present this data to the output.

A more detailed view of a memory architecture is shown in fig. 2.2. The *timing-and -coordination* block is split into a *read control* and *write control* block, and the sense amplifiers and bitline drivers are also separated. Additionally, the output of the sense amplifiers is fed back into the write drivers. Dynamic cells, which store data in the form of charge on a floating node, slowly lose their data due to

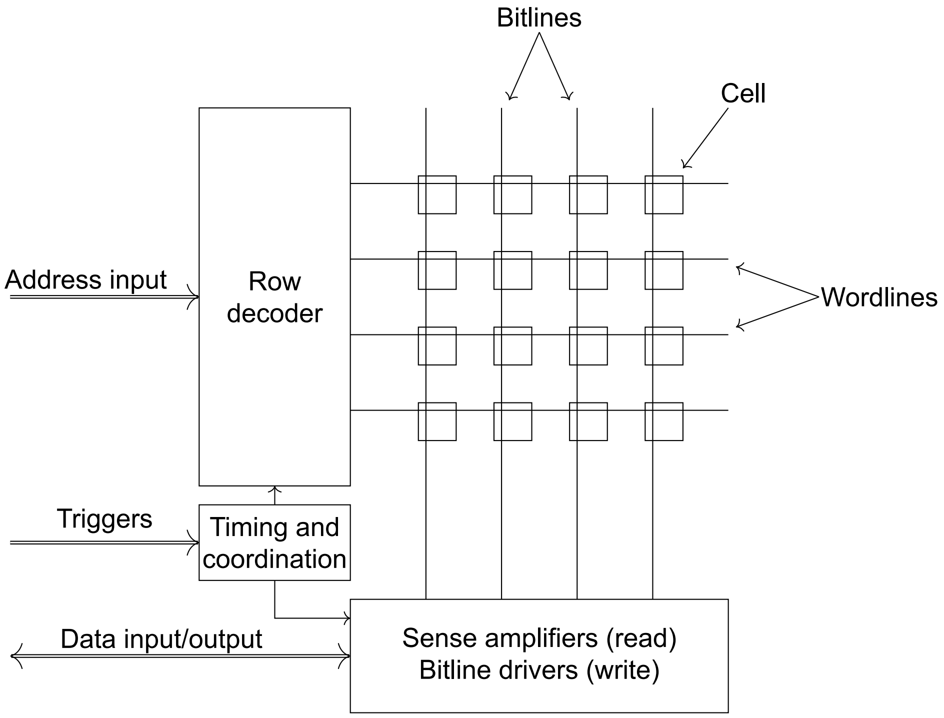


Figure 2.1: Basic memory architecture with a 4×4 cell array.

leakage and therefore require to be refreshed. This means that the cell is read and rewritten before the data is lost, refreshing the charge stored on the floating node. The direct connection from the sense amplifiers to the bitline drivers allows for refreshing without needing external hardware to ensure the data feedback. Finally, the sense amplifiers are followed by set of latches. They are used to isolate the sensitive sense amplifier circuits from the output of the memory and ensure that the data output always presents valid read data.

2.1.2. Memory cell designs

In this section, the schematics and operational details of four dynamic cell designs and one static cell design are explained.

1T1C dynamic cell

The 1T1C cell design is the most used dynamic memory cell due to its high integration density. It consists of a single pass transistor (1T) for access to a capacitor (1C) on which charge is stored, as shown in the schematic shown in fig. 2.3. With the use of special processes, such as trench capacitors, the capacitor can be implemented vertically with a capacitance in the order of $20\,\mathrm{fF}$ to $50\,\mathrm{fF}$ [16], [17], which results in tiny cells that can hold large amounts of charge.

Due to the special processes used, this cell design is rarely embedded in logic designs. Without these processes, a large cell area is required to get sufficient storage capacitance. It is therefore not considered as a viable cell design in the remainder of this thesis.

2T NW-PR dynamic cell

The first gain cell to cover is the dynamic 2T NW-PR memory cell. The cell name describes its schematic, which is shown in fig. 2.4, since it consists of two transistors (2T), uses an NMOS transistor for write operations (NW), and a PMOS transistor for read operation (PR). It is called a gain cell, because the gain of the readout PMOS is used to convert the storage-node voltage into a readout current. The main difference with the memory architecture described in the previous section, is the use of two wordlines (*RWL*/*WWL*) and two bitlines (*RBL*/*WBL*) per cell, one set for a write operation (W prefix) and one set for a read operation (R prefix). The general memory architecture still holds, but two row decoders are used, one for each set of wordlines (RWL and WWL), the sense amplifiers are

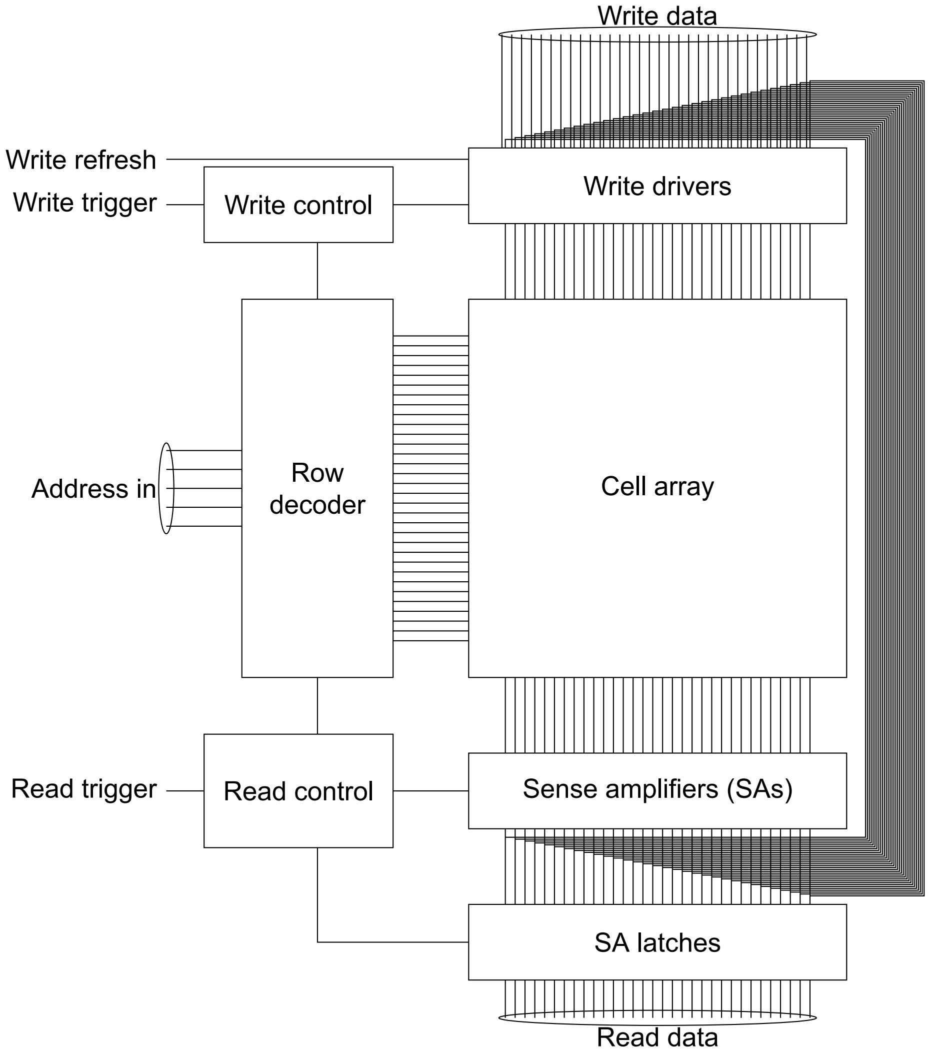


Figure 2.2: General memory architecture with a 32×32 cell array with refresh data shortcut.

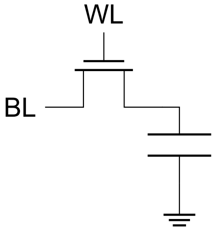


Figure 2.3: 1T1C dynamic memory cell schematic.

connected to the read bitlines (RBL), and the write drivers are connected to the write bitlines (WBL). In the following paragraphs, the three operation modes of this cell are covered.

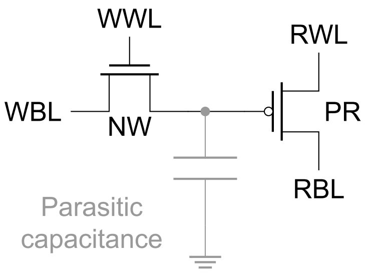


Figure 2.4: 2T NW-PR dynamic memory cell schematic.

Write To start a write operation, the WWL is pulled high which opens the NMOS pass transistor. If the WBL is driven low, the capacitance of the *storage node* (gate capacitance of PR + parasitic capacitance, in the order of $450\,\mathrm{aF}$) is discharged completely. If the WBL is driven high, the storage node is charged to roughly the WBL voltage minus the threshold voltage of NW. The write operation is ended by pulling the WWL low again. This causes the storage-node voltage to drop slightly due to the coupling between the WWL and the storage node. During the entire operation, the RWL and RBL are pulled low.

In this thesis, WWL boosting is not considered. By increasing the WWL beyond the nominal supply voltage, larger voltages can be written to the storage node. However, this could lead to premature device failure and requires additional design effort to ensure that voltages beyond the supply voltage can be handled.

Hold In the hold mode, the WWL, RBL, and RWL are pulled low. Ideally, this completely disconnects the storage node from the remainder of the circuit and the charge is maintained. However, if the WBL voltage is opposite of the stored voltage, sub-threshold (leakage) current through NW causes the storage-node voltage to drift, which weakens the stored state. Since the WBL is shared between different cells, this will be the case for roughly half the cells. This limits the total duration that the data is valid, also called the *retention time*. Other leakage sources include gate-oxide tunnelling or gate-induced drain lowering (GIDL) [18], reducing the retention time further.

Read In the read mode, the RWL is pulled high while the RBL has been precharged low. If the storage-node voltage is low, sufficient overdrive across PR will cause inversion and RBL will be charged. If the storage-node voltage is high, PR will remain in cutoff and the RBL will remain discharged. After a certain amount of time, the *read time*, the RBL voltage is compared with a reference voltage to determine the original cell state: 0 if the RBL voltage is larger than the reference voltage, and 1 if the RBL voltage is smaller than the reference voltage.

Due to the voltage drop across NW, the storage-node voltage can never reach the supply voltage. As a result, it may seem that the output transistor should always experience at least some inversion during readout since the difference between the storage node and the RWL (at the supply voltage) is about one threshold voltage. This is not the case due to the gate-to-source coupling capacitance of PR. When the RWL is charged, the storage-node voltage is pulled up, resulting in a gate-to-source voltage that is lower than the threshold voltage. This coupling voltage step also occurs if the storage-node voltage is low, which reduces the maximum overdrive that can be obtained and therefore limits the speed of the readout.

Due to the readout transistor coupling, the same transistor type 2T NW-NR and 2T PW-PR cell alternatives do not work without wordline boosting. For example, a PMOS write transistor would result in a storage-node voltage equal to the supply (high) or a single threshold voltage (low). Including the voltage increase on the storage node due to the coupling, the readout transistor will be fully off in case of a high storage-node voltage and either off or in very weak inversion in case of the low storage-node voltage. As a result, only small voltage differences will be generated on the RBL and therefore the cell is hard to read. A similar reasoning holds for the 2T NW-NR cell.

Using the *bitline voltage margin* method described in [19], the 2T NW-PR design is chosen over the 2T PW-NR design. In this method, the difference between the bitline voltages after a read operation for both states, the *bitline margin*, is simulated. For a similar time between the write and read operation, the *hold time*, the bitline margin of the 2T NW-PR cell is larger than that of the 2T PW-NR cell. This makes it easier for the sense amplifiers to distinguish the two states.

Finally, the maximum difference between the two bitline voltages is limited to roughly one threshold voltage. Unselected cells (RWL = 0) with a low storage-node voltage will cause a current from the RBL to the RWL when the RBL voltage exceeds the threshold voltage of PR. Besides limiting the maximum bitline voltage, it also increases the power consumption during read operations since the current is not used effectively. This puts an upper limit on the read time since the bitline voltage margin can only shrink once this limit has been reached and all additional energy is wasted.

3T NW-PR dynamic cell
A second dynamic gain cell, the 3T NW-PR cell, is shown in fig. 2.5. It is a three transistor cell (3T) with an NMOS write pass transistor (NW) and a PMOS read transistor stack (PR and PR'). It consists of a 2T NW-PR cell (NW and PR) with the RWL always pulled high (read mode), but the RBL charging current is gated by a second PMOS transistor (PR'). As mentioned before, the gate-to-source coupling of the readout transistor is the reason that the 2T NW-PR cell works. This coupling is not present in this design, so instead the threshold of PR should be increased to ensure that the transistor is fully off when the storage node has been written high.

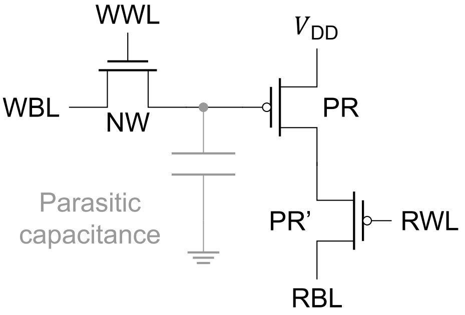


Figure 2.5: 3T NW-PR dynamic memory cell schematic.

This cell again has three operation modes which are similar to those of the 2T NW-PR cell design. The polarity of the RWL signal must be inverted, since it should be low when reading and high otherwise. The readout process is similar, where the RBL is charged towards the supply voltage if the storage-node voltage is low. Due to the additional gate, the readout leakage problem of the 2T NW-PR cell design is solved. Additionally, the overdrive on PR for a low storage-node voltage is larger since there is no coupling voltage step that increases the storage-node voltage during readout. This results in a faster readout than the 2T NW-PR cell.

3T PW-PR dynamic cell
The final dynamic gain cell is a 3T PW-PR cell with *preferential boosting* [20]. It is a three transistor cell (3T) with both a PMOS write transistor (PW) and a PMOS read transistor stack (PR and PR'). The schematic is shown in fig. 2.6 and is similar to the 3T NW-PR cell schematic.

The term *preferential boosting* comes from [20] and indicates the connection between the PR drain and the RWL. Similar to the 2T NW-PR transistor, this direct connection between the RWL and PR will cause a voltage step (down instead of up) on the storage node due to the coupling capacitance between the storage node and the RWL. It is called preferential since the step increases the difference between the storage-node voltages. If the storage-node voltage is high, there will be no channel and the coupling capacitance will be low. As a result, the voltage step down on the storage node is small. If the storage-node voltage is low, there is a channel in PR and the capacitance between the storage node and the RWL is higher. In this case, the voltage step down on the storage node is larger and therefore the storage-node voltage difference during readout is increased.

Contrary to the other cell designs with PMOS readout transistors, the RBL will be discharged for this design since RWL must be pulled low to open PR'. As a result, the RBL can not be discharged below

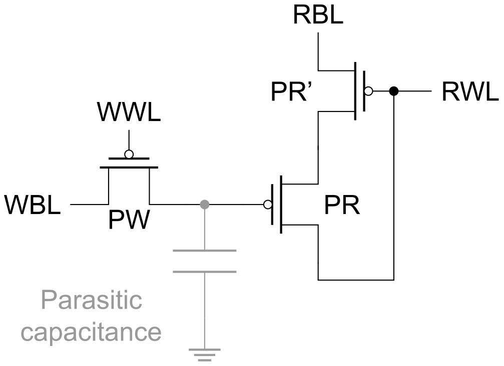


Figure 2.6: 3T PW-PR dynamic memory cell schematic.

a threshold voltage. This slows down the readout process for this cell type, but the step down on the storage-node voltage ensures sufficient overdrive voltage. Similar to the 3T NW-PR cell, this cell does not suffer from the readout leakage problem due to the pass gate in the readout stack. Contrary to the other dynamic cell designs, the readout of this cell is non-inverting: a high written voltage leads to a high bitline voltage, while a low written voltage leads to a low bitline voltage. This requires additional inversions throughout the full memory design, compared to the other dynamic cell designs, to ensure the correct data polarity.

6T static cell

The 6T static cell is the most used static memory cell and its schematic is shown in fig. 2.7. In this case, the data is stored in the state of a static bi-stable circuit consisting of two cross-coupled inverters. The cell does not require refreshing due to the static feedback, but is much larger than the dynamic cells since it requires six transistors (6T). The PMOS transistors are also called the *load* transistors (PL/PL'), the wordline transistors the *access* transistors (NA/NA'), and the inverter NMOS transistors the *driver* transistors (ND/ND'). It only has a single wordline (WL) for both read and write operations and uses a complementary bitline pair (BL/nBL) for both read and write operations instead of two separate bitlines. In the following paragraphs, the use of these lines in the three operation modes is explained.

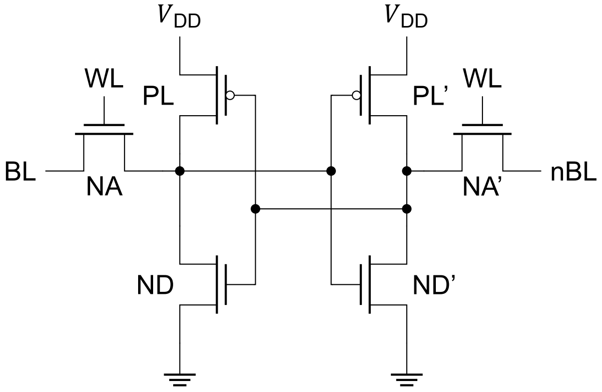
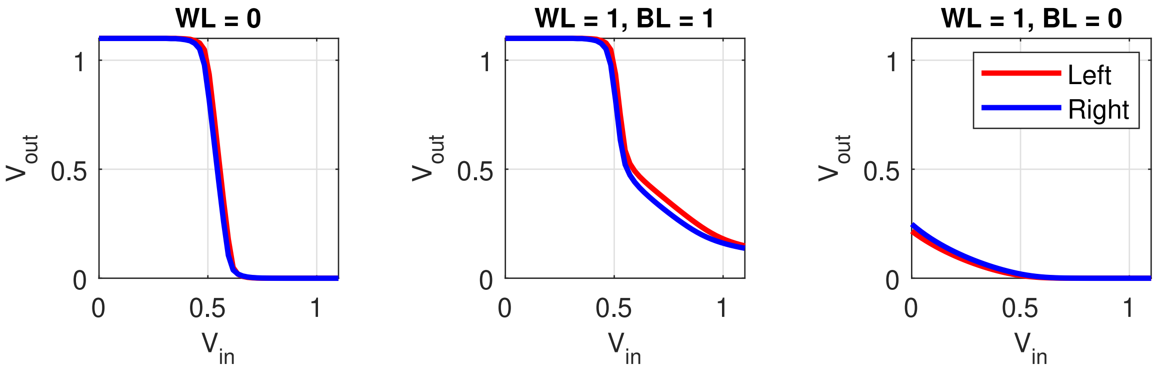


Figure 2.7: 6T static memory cell schematic.

Write During a write operation, the WL is high and the BL/nBL are driven with the complementary data to be stored. This causes the internal cell nodes to follow the bitlines through the access transistors and the cross-coupled inverters to latch, thereby copying the state from the bitline pair to the cell.

Read During a read operation, the WL is also high but the BL/nBL are not driven. Instead, they are precharged to the supply voltage and left floating during the read operation. The driver on the low side of the cell will pull down the connected bitline through the access transistor. Since the pull-down is



(a) WL = 0

(b) WL = 1 and BL = 1

(c) WL = 1 and BL = 0

Figure 2.8: Example transfer curves for the left and right inverter and pass transistor of a 6T static cell design, obtained from a two-point Monte Carlo simulation.

done through an NMOS with maximum overdrive, one of the bitlines is discharged extremely fast. By comparing the two bitlines, the low side can be determined and therefore the state can be retrieved.

Hold In the hold mode, the WL is low. This isolates the inverter pair from the bitline pair which means that no interaction takes place, neither write nor read, and the state is held by the inverter pair.

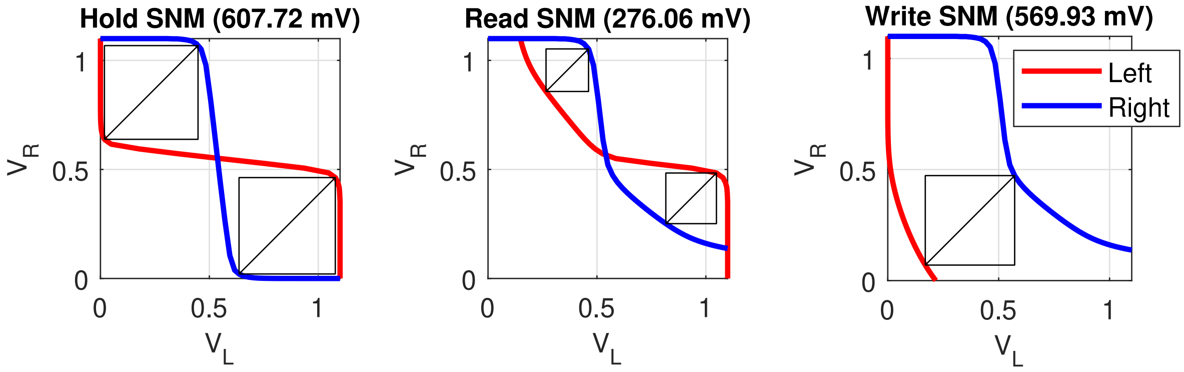
A tool often used in designing static memory cells is the *Static Noise Margin* (SNM) [21]. The SNM indicates the stability of the cell using two static (DC) transfer curves. These transfer curves are measured between the input and output of each inverter in three situations: with a low wordline (only the inverter), with a high wordline and a high bitline (inverter with a weak pull-up at the output), and with a high wordline and a low bitline (inverter with a strong pull-down at the output). This results in, for example, the transfer curves shown in fig. 2.8, obtained a two-point Monte Carlo simulation, for the left and right sections of the cell (PL/ND/NA and PL'/ND'/NA'). Ideally, they should be identical, but this is unlikely due to device mismatch.

The three transfer curves can be combined to find the SNM of the cell design in every operation mode. In the hold mode, the transfer curves from both sides with a low wordline are combined and plotted as shown in fig. 2.9a where the x- and y-axes indicate the two internal node voltages and the lines are the DC transfer curves of each inverter. The SNM is defined as the minimum of the two largest diagonal distances between these lines[1] and allows us to quantify the magnitude of a disturbance that is allowed until it would flip the cell. This margin should be positive and therefore requires that the lines cross, which means that there are two stable states.

During a read operation, both sides see a high wordline and initially a high bitline. These transfers can be combined and result in fig. 2.9b. The read SNM is again defined as the minimum of the largest differences between the curves and should be positive. This again means that two stable points are required, close to the two stable points in the hold mode. This ensures that a read operation does not disturb the state of the cell.

Similarly, in case of a write operation, one side sees a high wordline and low bitline, while the other side sees a high wordline and high bitline. A combination of these transfer curves is shown in fig. 2.9c. In this case, the smallest distance between the two curves should be maximised and the lines must not cross. As a result, there is only one stable point, which means that the cell can be forced into the desired state. For the shown example, BL is pulled low and nBL is pulled high such that the only stable state is the point where V_L is low and V_R is high. By flipping the high and low bitline sides, the figure should appear to mirror along the $x = y$ axis, and the single stable state must be on the other side of the graph where V_L is high and V_R is low. For this specific example, this gives a slightly larger margin and is therefore not shown.

[1] The side length of the square is sometimes used instead of the diagonal, resulting in a $\frac{\sqrt{2}}{2}$ scaling factor.



(a) Hold SNM of 607.72 mV.

(b) Read SNM of 276.06 mV.

(c) Write SNM of 569.93 mV.

Figure 2.9: SNMs for the example transfer curves shown in fig. 2.8 where V_L and V_R indicate the left and right node voltages of the cell, respectively.

2.2. Quantum computing

Quantum computing promises to solve certain problems in various fields that are intractable by classical computers using quantum algorithms [1]. A quantum computer is envisioned to be a heterogeneous system with a classical processor which loads quantum programs onto a quantum processor [22]. This is similar to the use of a modern general-purpose GPU. The quantum processor consists of a quantum device with qubits and a classical controller that performs the operations and readout on the quantum device by applying voltage and current waveforms. The quantum device and quantum computing in general are not the focus of this thesis and will therefore not be explained further. Instead, the focus is on the classical controller and the use cases for digital memories in them.

In this section, the use of digital memories in the classical controller for quantum computing is covered. Some proposed controllers are mentioned, their use of digital memory is listed and qualitative requirements for the memories are derived.

2.2.1. Classical controller architectures

Several classical controller architectures have been proposed for different qubit technologies and with various complexity [22]–[26]. The goal of these controllers is to provide an interface between software describing (mixed classical-)quantum algorithms and the quantum hardware. An example of such an architecture is shown in fig. 2.10. It consists of a classical pipeline for the classical instructions, which is used to perform computations with classical registers and control the program flow, and a quantum pipeline. The quantum pipeline is used to decode the quantum instructions, ensure that the timing of the instructions is correct, and apply the analog waveforms to the quantum device. The quantum pipeline can also perform readout of qubits on the quantum device and use the readout results to perform conditional quantum operations or return them to the classical pipeline.

2.2.2. Memory applications

Several memory applications can be found in the quantum processor part of the FT-QuMA architecture shown in the grey box in fig. 2.10. From left to right, the following memory applications are encountered:

- instruction and data memory,
- microcode control store,
- address registers,
- Q symbol table,
- logical measurement unit,
- timing control event queues,
- memory for the quantum error decoder (e.g. neural network [27], [28]),
- sequence memory,

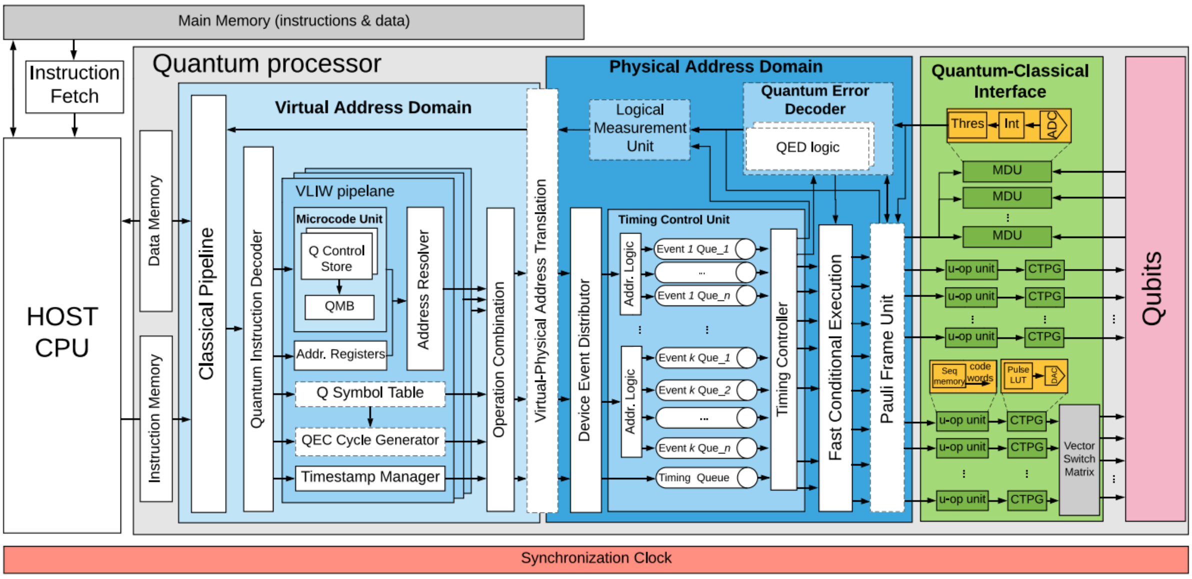


Figure 2.10: Architecture of the FT-QuMA controller architecture from [22, fig. 4].

- pulse lookup table (LUT).

A more detailed example of the memory use in the quantum-classical interface is shown in [29]. It uses three memories, roughly equivalent to an instruction memory that only stores the quantum instructions, a sequence memory, and a pulse LUT.

For each of the memories, some details about how they are operated can be derived. For example, the microcode control store, sequence memory, and pulse LUT are written at most once per program and read-only at various speeds during execution. While the microcode control store only needs to retrieve data once per instruction, pulse LUTs may require output data rates of $90\,\mathrm{Gbit\,s^{-1}}$ [29]. On the contrary, the timing control event queues will have an equal number of write and read operations. The remaining memories are both read and written during operation, but typically read more than written. Finally, the sizes may also vary significantly, since a sequence memory may need only a handful of words while the instruction memory may need to store programs of significant size.

The different use cases determine what memory type is most efficient at that position. A memory that is rarely used will benefit from a design that needs very little standby power, while a frequently used memory with mostly read operations requires a design with a low read operation energy and a high bandwidth. This trade-off is investigated in more detail in chapter 3.

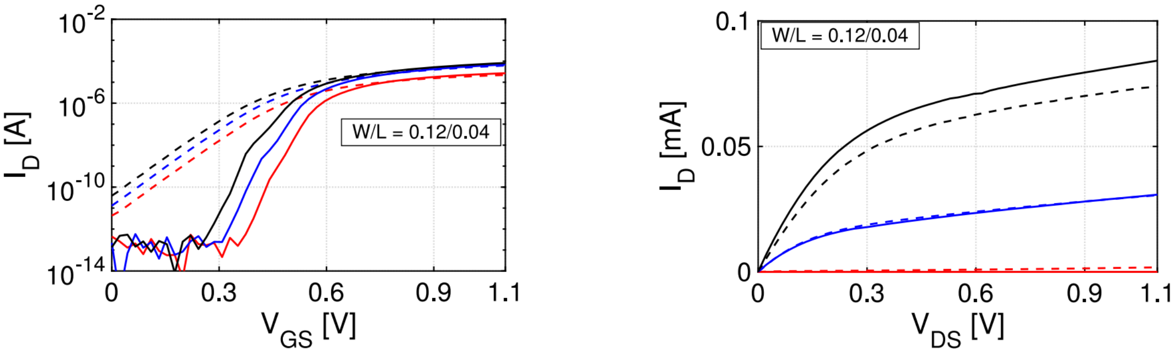
2.3. Cryogenic CMOS

To improve scalability, the classical controller should be located physically close to the qubits. Since the qubits require cryogenic operating temperatures, these control electronics should also be located at cryogenic temperatures [7], [8]. The qubits currently need to be cooled to temperatures of several mK, so dilution refrigerators are typically used. Since their cooling power is severely limited at these temperatures, the electronics are placed at a higher temperature plate around $4.2\,\mathrm{K}$. This also allows for testing of the electronics using a dipstick into liquid helium at $4.2\,\mathrm{K}$ instead of using a dilution refrigerator, which is much faster and cheaper. In either case, the delay, losses, and load of the long cables needed to get out of the fridge or dewar to room temperature to connect to equipment have to be taken into account when designing the hardware.

In this section, several effects of the temperature on the technology are summarised. This is followed by an overview of the state-of-the-art of digital memories at cryogenic temperatures and the observed effects of the cryogenic environment on them.

2.3.1. Effects on technology

The effects of cryogenic temperatures on various CMOS technologies have been shown in literature [11], [30]–[34]. A typical MOSFET transfer characteristic is shown in fig. 2.11 at both $300\,\mathrm{K}$ and $4.2\,\mathrm{K}$.



(a) Transfer characteristic where drain-to-source voltage = [0.1 V; 0.6 V; 1.1 V] [30, fig. 6c].

(b) Output characteristic where gate-to-source voltage = [0.43 V; 0.76 V; 1.1 V] [30, fig. 7c].

Figure 2.11: Measured transfer and output characteristics of thin-oxide NMOS in 40 nm CMOS with W/L = 120 nm/40 nm from [30]. Solid line: 4 K; dashed line: 300 K.

The main changes are as follows (exact values depend heavily on the technology and transistor sizing):

- The absolute threshold voltage of the devices increases roughly 100 mV to 200 mV due to scaling of the Fermi-Dirac distribution [35].
- The subthreshold slope becomes much steeper and decreases by roughly $60\,\mathrm{mV\,dec^{-1}}$.
- The carrier mobility increases by approximately a factor 2 due to a reduction of phonon scattering [36].

Additional effects that are not visible in fig. 2.11 but reported in literature are:

- The capacitance and leakage of the active-region-bulk junction diode decrease due to an increased depletion region width due to carrier freeze-out and therefore an increased barrier voltage [37].
- The mismatch between matched pair devices increases [33].
- The resistance of the metal interconnect decreases by 20 % to 50 % [38]–[41].
- The resistance of the substrate increases due to freeze-out and causes a current kink and hysteresis in larger technology nodes ($\leq 0.35\,\mu\mathrm{m}$) [30]. The current kink is not present in smaller technology nodes, but the increased substrate resistance may increase the latch-up risk [42].
- The channel noise decreases by a factor 10 due to the reduction in shot noise [38].

Combining some of the aforementioned changes results in the following functional differences. First of all, the leakage in the cutoff (subthreshold) region is significantly reduced both due to an increase in the threshold voltage and the steeper subthreshold slope. Second, the effect of the increase in threshold voltage on the output current at large overdrives is cancelled by the increase in carrier mobility, resulting in faster digital logic [30], [43]–[45].

2.3.2. Cryogenic digital memories

Digital memories at cryogenic temperatures have been topic of investigation in literature since 1979 [46]. There have mainly been two temperatures of interest, namely 77 K and 4.2 K. For speeding up regular computing, for example in data centres, 77 K is mainly of interest since it can be reached using liquid nitrogen which is cheap and easy to use. For use in lower temperature systems, such as quantum computing or superconducting computing, temperatures around 77 K are still used [47], but also 4.2 K is used since it can be reached using liquid helium [48]–[50].

In the remainder of this section, the changes in dynamic and static memories that have been reported in literature will be covered.

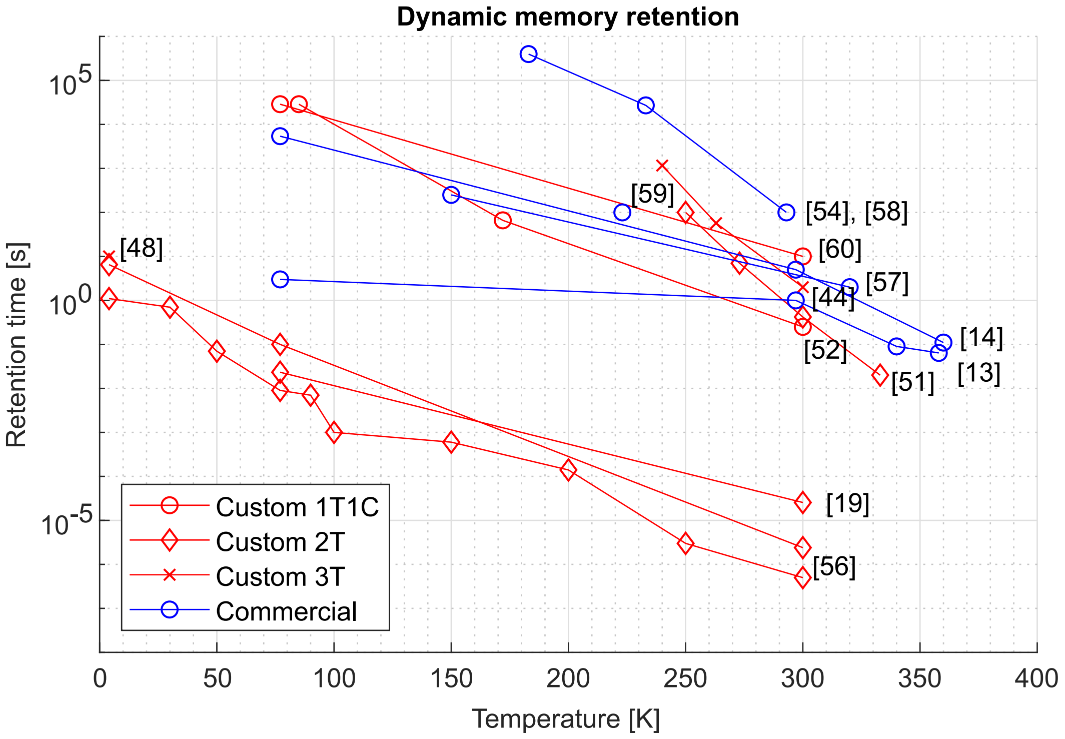


Figure 2.12: Retention time as function of temperature reported in literature. The red lines indicate various custom cell designs, while the blue lines indicate commercial DRAM products (all 1T1C).

Dynamic memories

Literature focusing on dynamic memories at cryogenic temperatures can be split into two groups: design and characterisation of a custom memory, and the characterisation of commercial DRAM sticks. Due to the limited data retention time of dynamic memories and the associated power consumption due to refresh operations, the focus has mainly been on determining the retention time of the dynamic cells.

Due to the reduced leakage, the retention time of dynamic cells increases significantly. Several works report long to near-infinite retention times and therefore a potential refresh rate of zero at $77\,\mathrm{K}$ [44], [46], [51], [52] and therefore at $4.2\,\mathrm{K}$ [50], [53], which could save in area as refresh circuitry is no longer needed. This seems to mostly be the case for large technology nodes at $0.35\,\mu\mathrm{m}$ or above, but the leakage at smaller nodes ($0.18\,\mu\mathrm{m}$ or smaller) can no longer be ignored [48]. Failure modes with low temperature dependence (low activation energy) such as tunnelling have been observed several times [13], [54], [55] and result in modern technologies in retention times in the order of several s [56]. Additionally, the effect of interference and disturb effects does not change and will require regular refreshes in order to protect the data [14], [57].

An overview of the retention times measured in literature can be seen in fig. 2.12. Due to the different ways retention time can be measured and the various technologies and cell types used, comparisons between different lines are not valid. However, a clear increase in retention time from $300\,\mathrm{K}$ to $4.2\,\mathrm{K}$ of roughly 10^6 can be seen, and most of the lines show approximately the same slope.

Apart from the increased retention time, an increase in speed has also been reported for both commercial DRAMs [61] and custom memories [56], [60], [62]. The access time decreases significantly for the unclocked memories (roughly halves), while the frequency of the clocked memories can be increased by $25\,\%$ to $30\,\%$.

Static memories

For static memories, the focus has mainly been on the increase in speed and static noise margins. An overview of the access time for static memory designs reported in literature is shown in fig. 2.13, with access time reductions between $20\,\%$ and $60\,\%$. Static noice margins have been reported to

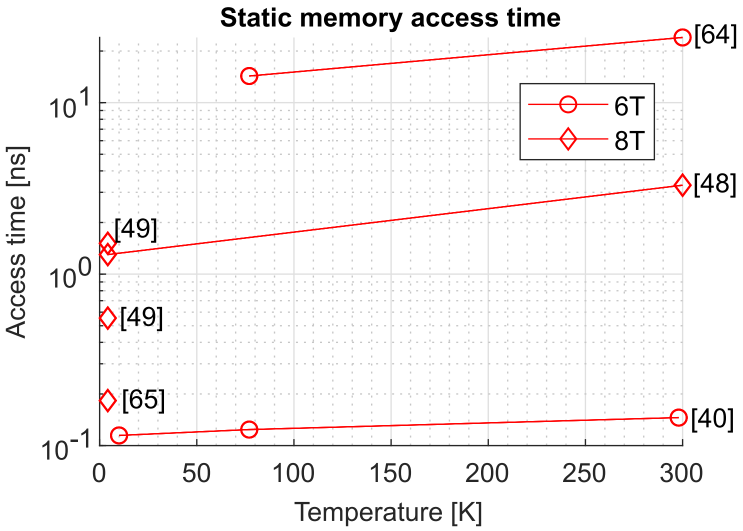


Figure 2.13: Static memory access time as function of the temperature reported in literature.

increase for low supply voltages due to the steeper subthreshold slope with threshold engineering [39] and by using write assist techniques [63]. Note however, that an SNM only shows static behaviour and that cells operating close to, or below, the threshold voltage may become very slow due to the steep subthreshold slope and increased mismatch.

3 Modelling

In this chapter, we will discuss the development of a model used to compare the performance and power consumption of the four memory cell designs at $4.2\,\mathrm{K}$ over a range of applications. Due to the lack of device models valid at this temperature, a memory model is first derived based on simulations at $-40\,^{\circ}\mathrm{C}$ ($233\,\mathrm{K}$) using mostly standard-threshold BSIM device models. At this temperature, some low-temperature effects such as reduced leakage are already visible. The resulting model is extended to $4.2\,\mathrm{K}$ by modifying it according to known cryogenic-CMOS effects and expectations based on previous work.

This chapter is structured as follows. We will first discuss how the memory application space is defined in section 3.1. In section 3.2, we will define the memory metrics of interest and show how they can be determined from simulation data by using the 2T NW-PR cell design as an example. The alternative cell designs will be treated in section 3.3 and the resulting memory metrics will be compared. In section 3.4 we will adapt the developed model to predict memory performance at $4.2\,\mathrm{K}$ and see the effects on the optimal memory design for various applications. Finally, we will reflect on the model, identify limitations, and propose improvements in section 3.5 and conclude the chapter in section 3.6.

3.1. Applications

In this section, we will investigate what is meant with a memory application and define an application space over which the memory designs will be evaluated.

A set of memory application requirements tells us how we want to use a memory and what the restrictions on the design are. This includes many aspects, leading to a huge design space. Some of these aspects can be derived from how we intend to use the memory and will largely determine the architecture. This includes, for example, the data access pattern (random or sequential), desired throughput, size of the array, size of a word/row, maximum latency, average read/write frequency, error tolerance, etc. Additionally, there are some aspects that limit certain metrics of a memory design, such as the area and power consumption. For each application, a different design is possible using architectural techniques. One of these is banking, where we access multiple arrays at the same time to increase the throughput, especially for sequential access patterns. The resulting design space is too large and specific for easy comparison.

Since our focus lies on memory cell design, we will use only a single architecture design. To keep the design simple, we will consider a single memory bank of $1\,\mathrm{Kibit}$ ($1024\,\mathrm{bit}$) with 32 words of $32\,\mathrm{bit}$ each, which is equal to 32 rows and 32 columns. Additionally, we will assume a random access pattern due to its versatility. This allows us to compare the performance of the memories based on the choice of cells without taking architectural differences into account. This also means that the possible memory application requirements space shrinks significantly since several requirements are now fixed or directly dependent on each other. For example, the throughput is now inversely proportional to the latency since the number of banks is fixed.

After reducing the application and design spaces, we are left with six application aspects. Two of these directly relate to how the memory is used, namely the *average read and write frequency*. We will use these values to define an application space. The remaining four put restrictions on several memory metrics. They will limit the allowed *area*, *latency*, *error rate*, and *power consumption* of the memories and can be used to pick the memory design that delivers the best trade-off.

3.1.1. Application space

As explained before, the application space for this model has been reduced to only the frequency of read and write operations. Since the architecture of the different memories is the same, the application can only determine how we use it. Using the average number of read and write operations per second thus allows us to describe many different applications.

We can limit one dimension of the application space by replacing the write frequency with the average number of write operations per read operation. Since the read and write frequencies can differ significantly, the application space stays large. Note however, that the average write frequency is usually lower than the average read frequency[1]. We can therefore define the application space using an average read frequency, which can span many orders of magnitude, and a ratio of write operations per read operation, which is limited between 0 and 1.

The defined application space can be used to classify the applications mentioned in section 2.2.2 as shown in fig. 3.1. For example, a queue, such as the timing control event queues, will have a write/read operation ratio of 1, while a read-only memory, such as a LUT, has a write/read operation ratio of 0. Finally, working memory for a processor or controller will have a write/read operation ratio between 0 and 1, but exact values depend heavily on the program. The application of a memory then also defines whether is can be slow or needs to be fast. For example, a LUT storing bias voltages that need to be refreshed at low rates can be slow [66], while a pulse LUT for an arbitrary waveform generator (AWG) needs to be fast. The throughput of various memories in a hierarchical memory organisation can also vary several orders of magnitude [29].

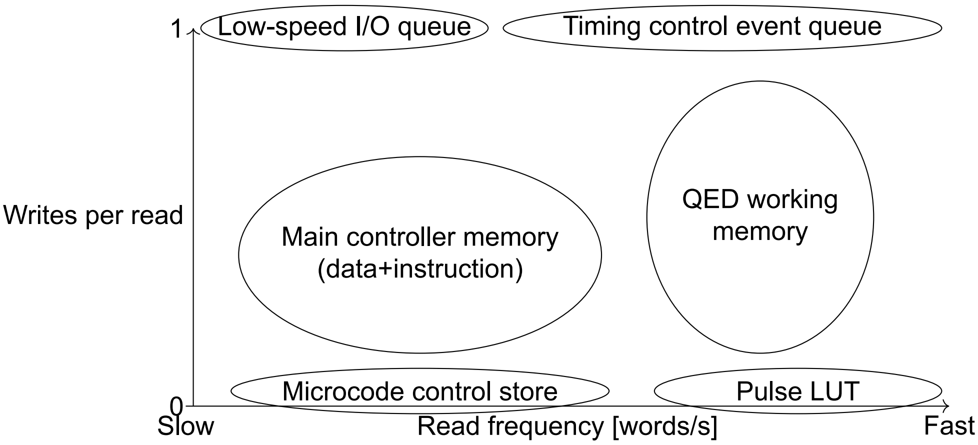


Figure 3.1: Memory applications in the application space defined by the read frequency and the average number of write operations per read operation.

3.2. Memory metrics

In the previous section, we have seen how an application space was derived from the memory usage requirements. This leaves us with the restrictions on four remaining memory metrics: the area, latency, bit error rate, and power consumption. In this section, we will show how we can compute these metrics for the different memory designs and how they are related.

We will go through the memory metrics in the following order by using the 2T NW-PR cell type as an example. The schematic and operational details of this cell design can be found in section 2.1.2. We will first look at the area of the design, followed by its latency. Next, we will see how we can determine the Bit Error Rate (BER) and how it is related to the latency. Finally, we can calculate the power consumption of the design over the application space and see how all the metrics are related and their trade-offs.

[1] Writing data that is never read is a waste of resources, but may happen in certain applications such as caches.

3.2.1. Area

Of all the memory metrics, the area is the easiest is to calculate. Nevertheless, it is important to consider since memories can get extremely large and can be responsible for more than half of the area consumption of a processor [67].

To calculate the total area, we can add the area of the cells and the peripherals. The total area of the cells can be estimated using eq. (3.1) where h_{cell} and w_{cell} indicate the height and width of the cell, while N_{cells}, N_{rows}, and N_{columns} indicate the total number of cells, rows, and columns, respectively.

$$A_{\text{cells}} = N_{\text{cells}} \times A_{\text{cell}} = (N_{\text{rows}} \times N_{\text{columns}}) \times (h_{\text{cell}} \times w_{\text{cell}}) \tag{3.1}$$

The area of the peripherals can be estimated using eq. (3.2) where w_{peri} and h_{peri} indicate the width and height of the peripheral overhead. The row peripherals include the row decoders, while the column peripherals include bitline drivers, sense amplifiers, and data latches.

$$\begin{aligned} A_{\text{peri,rows}} &= N_{\text{rows}} \times h_{\text{cell}} \times w_{\text{peri}} \\ A_{\text{peri,columns}} &= N_{\text{columns}} \times w_{\text{cell}} \times h_{\text{peri}} \\ A_{\text{peri}} &= A_{\text{peri,rows}} + A_{\text{peri,columns}} \end{aligned} \tag{3.2}$$

A visual representation of the aforementioned areas can be seen in fig. 3.2. Note that this is a rough estimation of the area, taking only the cells, row decoders, and bitline stacks into account. Additional area will be required for the timing-and-coordination circuits and dummy structures to ensure cell matching.

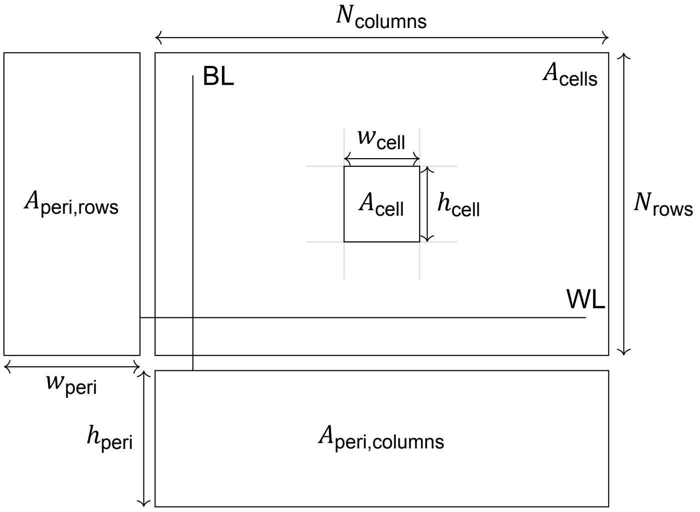


Figure 3.2: Simplified memory layout used for area estimation.

For the 2T NW-PR memory cell this leads to the following area estimation. Based on a preliminary layout schematically shown in fig. 3.3, we can estimate the dimensions in fig. 3.2, as reported in table 3.1. This leads to a cell area of $0.2268\,\mu\text{m}^2$ and a total area for the array of $466\,\mu\text{m}^2$ (for $N_{\text{columns}} = 32$ and $N_{\text{rows}} = 32$).

Table 3.1: 2T NW-PR array dimension estimations

Dimension	w_{cell}	h_{cell}	w_{peri}	h_{peri}
μm	0.63	0.36	8	7

Finally, it is instructive to observe some trends for the area metric. Increasing the total number of cells will linearly scale the total cell area, while scaling the peripheral area sublinear. This means that especially for larger arrays, the total cell area becomes the dominant area consumer. In the previously shown situation with 32 words of 32 bit, the cell area is only 41 % of the total area, but scaling the array to 64 words of 64 bit would result in a cell area of 59 % of the total area. It is therefore worthwhile to minimise the area of a single cell.

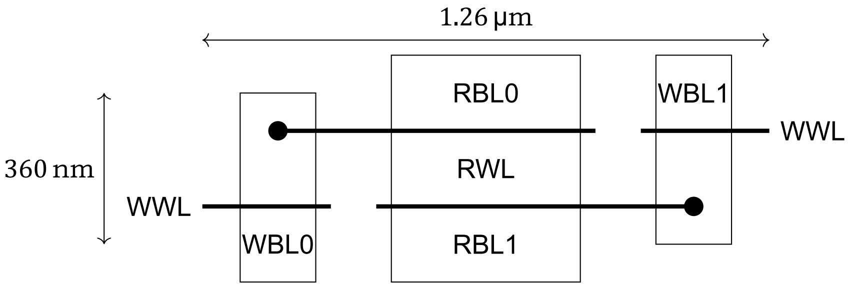


Figure 3.3: Simplified layout of a double 2T NW-PR cell design with the two write transistors (NW) on the outsides and the readout transistors (PR) stacked in the middle. These two cells share the same wordlines, but have different bitlines, which means that they are in the same row but different column. To obtain the area of a single cell, the width therefore has to be divided by 2.

3.2.2. Latency

The latency of a memory is defined as the time between the start of a memory operation and its completion. We will separately consider the write and read latency for write and read operations, respectively, in the remainder of this section. However, to simplify interaction with the memory, a single *combined memory operation latency* is defined as the largest of the two.

The duration of both operations is the result of the timing of the peripherals by the control circuits. It is therefore not an inherent property of the cell design itself, but the cell design must be taken into account to determine the optimal timing. This optimal timing is the result of a trade-off between the latency and other memory metrics, which is investigated in more detail in section 3.2.5.

Write latency

The write latency is mainly of interest for dynamic cells. The storage-node voltage of dynamic cells is written through a pass transistor. Depending on the type of pass transistor, one voltage level is passed strongly while another is passed weakly. For the 2T NW-PR cell, the N-type write pass transistor will strongly pass a low voltage ($0\,\mathrm{V}$) and weakly pass a high voltage ($1.1\,\mathrm{V}$). In the latter case, if the storage-node voltage starts at $0\,\mathrm{V}$, the storage node will be initially charged quickly. However, as the storage-node voltage increases, the transistor current will decrease. This will lead to a slow settling towards a storage-node voltage of $1.1 - V_{\mathrm{th}}$.

Figure 3.4 shows Monte Carlo simulation results of the settling of the storage-node voltage of a 2T NW-PR cell. The cell is initialised with $0\,\mathrm{V}$ or $1.1\,\mathrm{V}$, the write bitline is connected to the opposite voltage, and the write wordline is pulled high. The red lines shows settling of a strong write, while the blue lines show settling of a weak write. The black lines show the $\pm 3.3\sigma$ spread, assuming a normal distribution at every time instance. The probability of being outside this range is $\frac{1}{1024}$, so one out of the 1024 cells is expected to lie outside this spread.

We can clearly see the difference between the strong and weak write. The strong state (red) is written completely within $10\,\mathrm{ps}$. The weak state (blue) starts of as fast as the strong state, but quickly slows down due to the reducing transistor overdrive and ends up settling exponentially slower (note the logarithmic x-axis in fig. 3.4). Writing for longer will increase the difference in storage-node voltage between the two states and therefore lead to easier state detection when read. However, due to the slow settling the advantages quickly wear off. Therefore, setting the write duration equal to the read duration gives the best results as it gives the largest possible storage-node voltage difference without slowing down the combined operation latency.

For static cells, once a cell has flipped, writing longer does not give any advantage. This means that we have a strict lower bound for the duration of a write operation for a static cell to guarantee that it can be flipped, but going beyond it gives no advantage nor disadvantage as long as it does not slow down the combined operation latency.

For this model, we will fix the duration of a write instruction to $1\,\mathrm{ns}$. This means that in all the simulations, we will perform write operations by opening the write pass gate (NW in fig. 2.4) for $1\,\mathrm{ns}$. This is sufficient for a strong write as it is longer than $10\,\mathrm{ps}$. For the weak write, it puts us around $80\,\mathrm{mV}$ into the settling region which starts around $100\,\mathrm{ps}$ at $700\,\mathrm{mV}$. This will result in a total write operation latency that is slightly longer than $1\,\mathrm{ns}$ due to peripheral overhead. It is also similar to the read latency which we will look at next, which means that we get the longest possible write time without limiting the

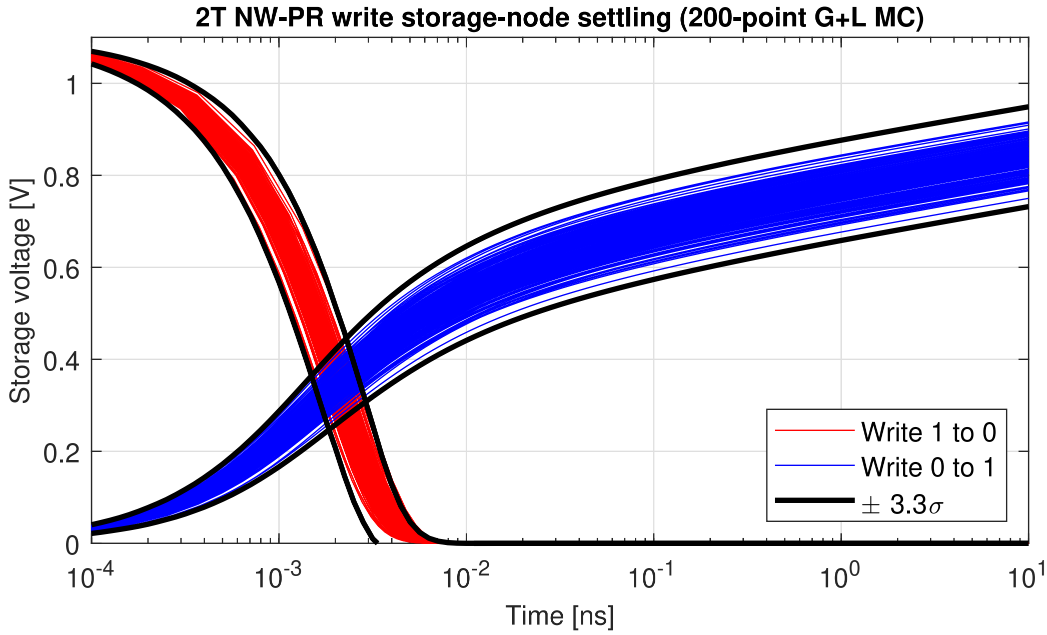


Figure 3.4: 200-point Monte Carlo (global + local) simulation of the settling of the 2T NW-PR storage-node voltage during a write operation. Black lines indicate $\pm 3.3\sigma$ ($\frac{1}{1024}$) for every time instance of the two curve families.

combined operation latency.

Read latency

The required duration of a read operation depends on many factors. During a read operation a bitline is (dis)charged, the speed of which is determined by the cell strength and the total bitline capacitance. The read duration then determines the difference between the bitline voltages for the different cell states.

Figure 3.5 shows the bitline voltage for a 2T NW-PR cell during a read operation for various read durations from $100\,\mathrm{ps}$ to $1\,\mathrm{ns}$. The shown process is as follows. First, the bitline is precharged to ground. Next, the read wordline is pulsed for $100\,\mathrm{ps}$, $250\,\mathrm{ps}$, $400\,\mathrm{ps}$, $550\,\mathrm{ps}$, $700\,\mathrm{ps}$, $850\,\mathrm{ps}$ and $1000\,\mathrm{ps}$. Finally, $500\,\mathrm{ps}$ after the falling edge of the read wordline, the bitlines are precharged to ground again. Reading for longer gives a larger bitline voltage difference between the two states, or 'bitline margin', since we follow the bitline charging trajectory for longer.

For dynamic cells, these bitline voltages not only depend on the read operation, but also on the time between successive writes and reads, or *hold time* ($\mathrm{t_{hold}}$). Between the write operation and the read operation, the storage-node voltage will increase or decrease due to leakage for the low or high writes, respectively. This will reduce the overdrive in case of a write-0, leading to a lower charging current. In case of a write-1, this eventually leads to inversion which increases the charging current. This can also be seen in the right plot of fig. 3.5, where the bitline is charged slower than in the left plot after a write-0. Additionally, the bitline is charged slowly after a write-1, which leads to a smaller bitline margin. However, the bitline margin is still increasing for an increasing read duration.

The left figure of fig. 3.5 shows the readout leakage that the 2T NW-PR cell design suffers from. Once the bitline voltage reaches approximately $0.52\,\mathrm{V}$, there is a balance between the current supplied by the cell that is read and the leakage of other cells that have been written low. As soon as the read wordline is pulled down, the current from the cells that have been written low discharges the bitline again towards roughly $0.4\,\mathrm{V}$.

It may seem that reading longer will result in a better bitline margin, but this is not the case. In fig. 3.6, we can see that the maximum bitline margin is achieved for a read time of roughly $2.5\,\mathrm{ns}$ for a hold time of $40\,\mathrm{\mu s}$. The initial increase in bitline margin is due to the bitline being charged faster in case of a write-0 than a write-1, as shown in fig. 3.5. However, once the bitline voltage for the write-0 starts to settle due to the readout leakage, the bitline voltage for the write-1 still increases, lowering the bitline margin again. This long hold time will however result in a relatively slow memory with a high read energy consumption for short hold times due to readout leakage. For example, for a hold time

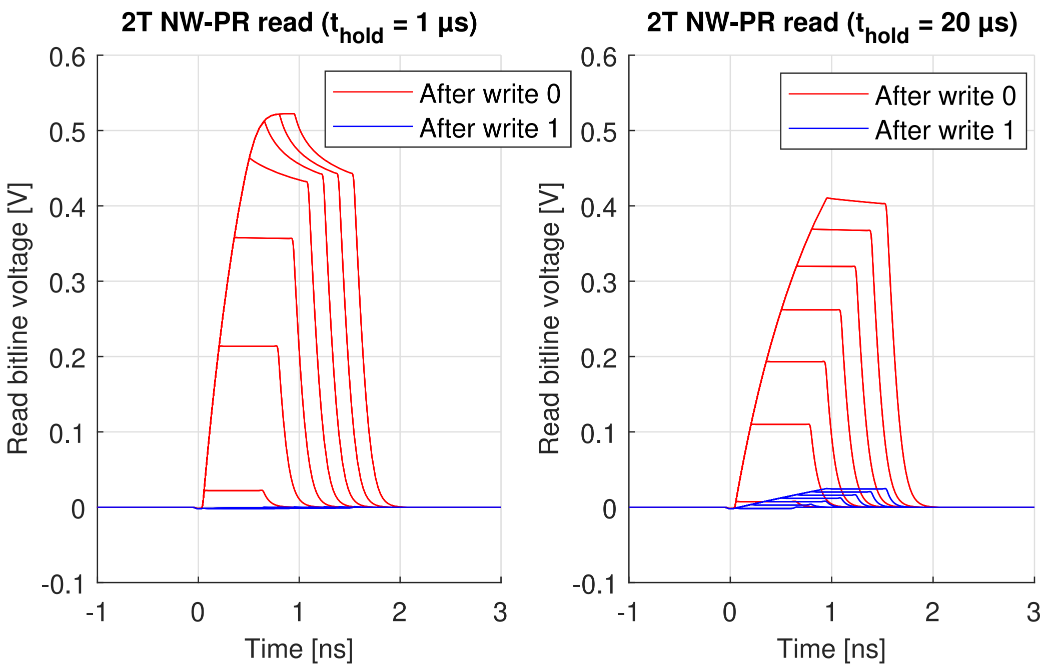


Figure 3.5: 2T NW-PR read bitlines during a read operation with durations of 100 ps, 250 ps, 400 ps, 550 ps, 700 ps, 850 ps and 1000 ps. Left: read 1 µs after write, right: read 20 µs after write.

of 1 µs, the leakage current with a maximum of almost 11 µA starts to flow within 500 ps, resulting in an additional energy loss of over 22 fJ (> 10 % of the total read power, as calculated in section 3.2.4). In order to reduce this leakage, keep the latency low, and maintain sufficient bitline margin for various hold times, the read time is fixed to 1 ns.

Whatever the read time, the bitline margin of dynamic cells will collapse over time. This happens due to leakage, which brings the storage node voltages of the different states closer together. As a result, the (dis)charging current of the readout transistors for the different states get closer together. At a certain point in time, the two states can no longer be reliably distinguished. To determine when this happens, we need to take a look at the bit error rate.

3.2.3. BER

To determine when we need to refresh our memory, we need to know when the bit error probability of a read operation becomes too large. This is where the *Bit Error Rate* (BER) comes in. First, we will see how a noiseless yield can be computed. This is followed by a BER analysis to find the actual expected error rate including noise.

Noiseless yield

The *yield* metric will give use the probability of a good cell, where a good cell will always be read correctly within the refresh period in the absence of noise. To calculate this, we need mismatch information about the cells and sense amplifiers, and a reference against which the cell voltages are compared.

Cell mismatch To find mismatch information about the cell, Monte Carlo simulations of the read operation for various hold times are used. This will result in distributions of the bitline voltages for the two states over hold time. We should be able to determine, for a given hold time, the data state of the cell from the observed read bitline voltage.

The left graph of fig. 3.7 shows the bitline curves for a 200 point Monte-Carlo simulation of the 2T NW-PR cell. If we take a vertical slice from this figure for a given hold time, we can fit the voltages to a distribution. The bitline voltages that belong to the 0-state best fit to a normal distribution, while the bitline voltages that belong the 1-state best fit to a log-normal distribution. This makes sense as the readout transistor is in strong inversion during a read of the 0-state, since the read wordline voltage is high while the storage-node voltage is low. This causes the readout current to be roughly proportional to

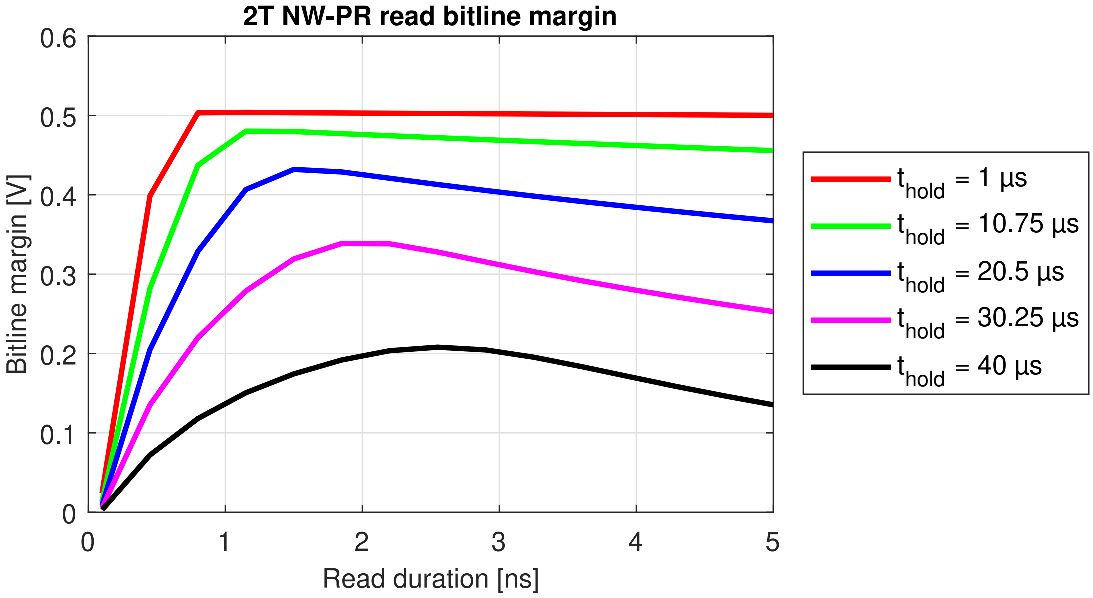


Figure 3.6: 2T NW-PR read bitline margin as a function of the read duration for various hold times.

the threshold and therefore the read bitline voltage mismatch to be proportional to the device mismatch. During a read of the 1-state, the readout transistor is in the subthreshold operating region, since both the read wordline voltage and the storage-node voltage are high. This causes the readout current to be exponentially dependent on the threshold and therefore the read bitline voltage mismatch to be exponentially proportional to the device mismatch. For example, at a hold time of $20\,\mu\text{s}$, the bitline voltages follow the distributions shown in eq. (3.3) and in the right graph of fig. 3.7.

$$\begin{aligned} V_0 &\sim N(\mu,\sigma^2), \text{ where } \mu = 0.4057 \text{ and } \sigma = 0.04329 \\ V_1 &\sim \ln N(\mu,\sigma^2), \text{ where } \mu = -3.475 \text{ and } \sigma = 0.3721 \end{aligned} \tag{3.3}$$

Sense amplifier mismatch Mismatch information about the sense amplifier is harder to determine as it highly depends on the sense amplifier design. Based on simulations of a preliminary design, we find that the input-referred offset follows a normal distribution with zero mean and a standard deviation of roughly $16.5\,\text{mV}$. The exact value can always be adjusted based on the final design, so this value is assumed for all memories for now.

Combining the statistics For the yield, the probability that a random cell and sense amplifier with a specified reference voltage result in an incorrect read after a certain hold time is computed. The algorithm to do this is as follows:

1. *Obtain the distributions of the bitline levels at the given hold time from the cell mismatch information.* This is done by taking a vertical slice of the bitline graph and fitting the bitline voltages to a normal and log-normal distribution. An example of this is shown in fig. 3.7 with the 2T NW-PR cell for a hold time of $20\,\mu\text{s}$.

2. *Find the probability that a perfect decision against any reference is correct.* In this case, the bitline voltage distributions for the cell are assumed to be independent and any combination for the two states is assumed equally likely. The error probabilities for the two states from the distributions obtained in the first step is computed. Since the states are assumed to be equally likely, the combined error probability is given by the average error probability. This is shown in the left graph in fig. 3.8 for the example distributions from step 1 (eq. (3.3)).

3. *Multiply the perfect decision error probability density function with the reference probability density function.* The reference probability distribution follows a normal distribution with its mean

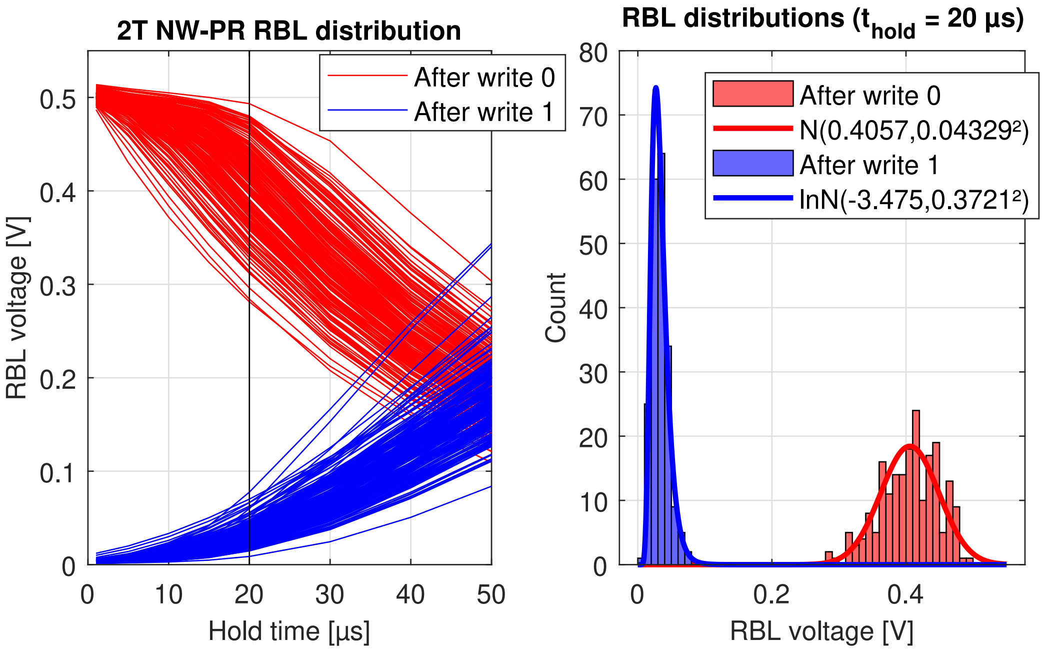


Figure 3.7: Left: 200-point Monte Carlo simulation (tt corner + local variation at $233\,\mathrm{K}$) of the read bitline voltages for 2T NW-PR cells for various hold times. Right: distribution of read bitline voltages for a hold time of $20\,\mu\mathrm{s}$ (black line in the left figure).

equal to the specified reference voltage and its standard deviation equal to the sense amplifier mismatch. For the example, a threshold voltage of $200\,\mathrm{mV}$ and the earlier determined sense amplifier mismatch of $16.5\,\mathrm{mV}$ are assumed. The resulting probability density function is shown in the right graph of fig. 3.8 and gives the combined probability of achieving a certain effective threshold voltage (applied threshold + sense amplifier offset) and that threshold leading to an incorrect read.

4. *Integrate the resulting probability density function to find the total probability of an incorrect read.* For the given example, this leads to a total read error probability of 2.514×10^{-6}. This means that, without noise, reading a random cell with a random sense amplifier and a reference voltage of $200\,\mathrm{mV}$ after $20\,\mu\mathrm{s}$ has a probability of 2.514×10^{-6} of resulting in the wrong data, assuming equal write data probability.

The previously described method to find the yield can now be applied for a range of hold times and reference voltages. The resulting values for each combination can be put into a graph, resulting in fig. 3.9, where the black lines indicate constant yield values equal to $1 - 10^{-x}$ and the red line shows the reference voltage that gives the best yield for a given hold time. Given a desired yield, we can pick the optimal reference voltage that gives us the longest retention time. For example, if we pick a yield of $1 - 10^{-6}$, we find an optimal reference of roughly $190\,\mathrm{mV}$ (actually $186.8\,\mathrm{mV}$) and a maximum hold time of $19.64\,\mu\mathrm{s}$. This means that the refresh rate must be $50.91\,\mathrm{kHz}$ to guarantee this yield for all read operations.

A yield of $1 - 10^{-6}$ for $1\,\mathrm{Kibit}$ is plenty and the probability of getting a bad cell is very small, roughly one per 489 memories. However, some cells may be considered good cells in the absence of noise, but result in high error rates once noise is taken into account.

Noisy BER

To determine the BER including noise, a Monte Carlo simulation is used. In this simulation, a large number of memory cores are simulated with 1024 cells and 32 sense amplifiers, randomly selected from their respective distributions. For the cells, a random read bitline voltage for both data states is drawn from the distribution at a hold time of $20\,\mu\mathrm{s}$ (eq. (3.3)), assuming the two distributions are independent. For the sense amplifiers, a threshold that includes a random input-referred offset is selected from a normal distribution around the chosen threshold of $190\,\mathrm{mV}$ with an input-referred offset standard deviation of $16.5\,\mathrm{mV}$. For each cell and associated sense amplifier, the probability of an incorrect read

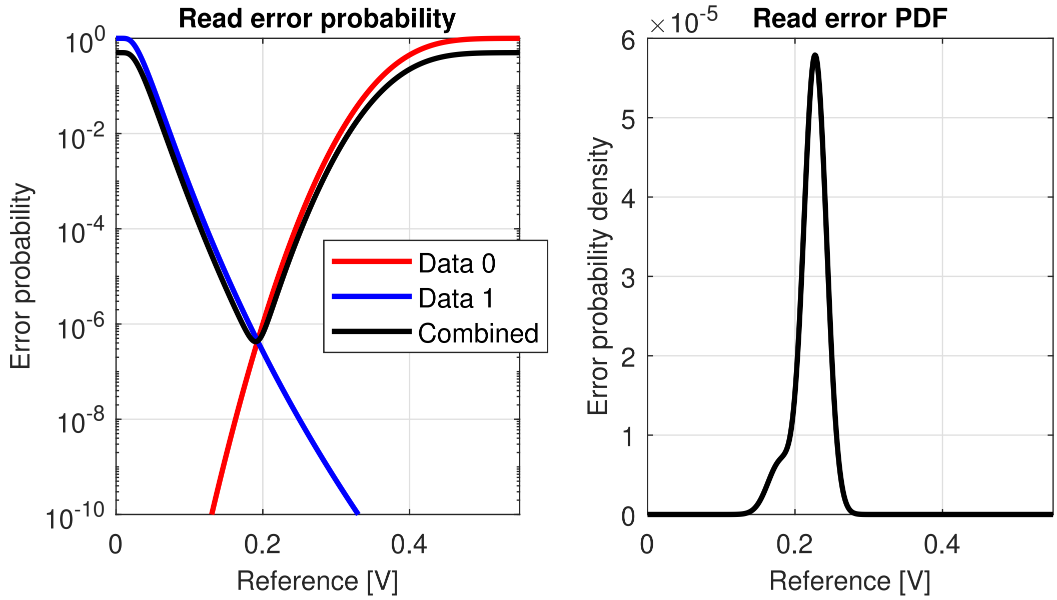


Figure 3.8: Left: probability of incorrect read of random cell with given reference voltage after a hold time of $20\,\mu\mathrm{s}$. Right: combined probability density of the effective reference level seen by a sense amplifier due to input-referred offset and that leading to an incorrect read with equal data probability.

decision is determined for both data states, including sense amplifier input-referred noise following a normal distribution with a standard deviation of $10\,\mathrm{mV}$. In reality, the noise will be even smaller, in the order of $4\,\mathrm{mV}$. However, $10\,\mathrm{mV}$ gives a reasonable idea of the cell BER distribution that can be expected and does not lead to numerical overflow[2]. Finally, the read error probabilities for both states are averaged, assuming equal data probability, to find the total read error probability.

A histogram with the read error probability for all cells of $2^{16} = 65536$ memory cores with 32 words of $32\,\mathrm{bit}$ (2^{26} cells in total) is shown in the left figure of fig. 3.10a and appears to follow a log-normal distribution[3]. The median cell has a BER of 3.89×10^{-54} and there is a probability of $99.9\,\%$ of getting a BER lower than 10^{-12}. The worst performing cell of every memory is shown in the right figure of fig. 3.10a. This distribution also approximately follows a log-normal distribution, and means that in half the memories the BER for every cell is smaller than 8.72×10^{-11}, but the probability of getting a worst cell BER lower than 10^{-6} is only $85.0\,\%$.

The computed BER only holds for read operations performed $20\,\mu\mathrm{s}$ after a write operation. This BER therefore limits the quality of the refresh read operation, as earlier reads will see much better bitline margins (for a hold time of $10\,\mu\mathrm{s}$, the worst cell BER is higher than 6.22×10^{-13} in $99.99\,\%$ of the memories). It is therefore instructive to see how long it would take, on average, for a bit to flip due to an incorrect refresh read operation. With a refresh rate of $50\,\mathrm{kHz}$, a worst cell BER of 10^{-6} results in one expected bit flip every $20\,\mathrm{s}$ due to refresh read operations alone. Note that this is only on the worst cell in the array. In half of the memories, the BER of the second worst cell is already smaller than 4.82×10^{-14}, and the probability of getting a second worst cell BER lower than 10^{-8} is roughly $95.0\,\%$. With a refresh rate of $50\,\mathrm{kHz}$, a BER of 10^{-8} results in an expected bit flip every $33.3\,\mathrm{min}$. The cell performance quickly improves, but the number of errors is still high due to the large input-referred noise of the sense amplifiers.

The BER becomes a lot better for decreasing amounts of noise. If we decrease the input-referred noise standard deviation to $8\,\mathrm{mV}$, the median cell BER becomes 3.13×10^{-83} and $99.98\,\%$ of the cells have a BER below 10^{-9}. Additionally, the median worst cell BER becomes 1.34×10^{-15}, which corresponds with an expected refresh bit flip every $473\,\mathrm{yr}$. For $83.9\,\%$ of the memories, the worst cell BER is smaller than 10^{-9}, or one expected refresh bit flip every $5.55\,\mathrm{h}$. The BER histograms for an

[2] With input referred noise standard deviations below $7\,\mathrm{mV}$, the BER of some cells lies outside of the IEEE 754 Double-Precision Floating-Point range ($< 4.94 \times 10^{-324}$), causing issues with distribution fitting.

[3] The BER has a limited support between 0 and 1, while a log-normal distribution has infinite support starting from 0. Nevertheless, the distribution shapes match very well.

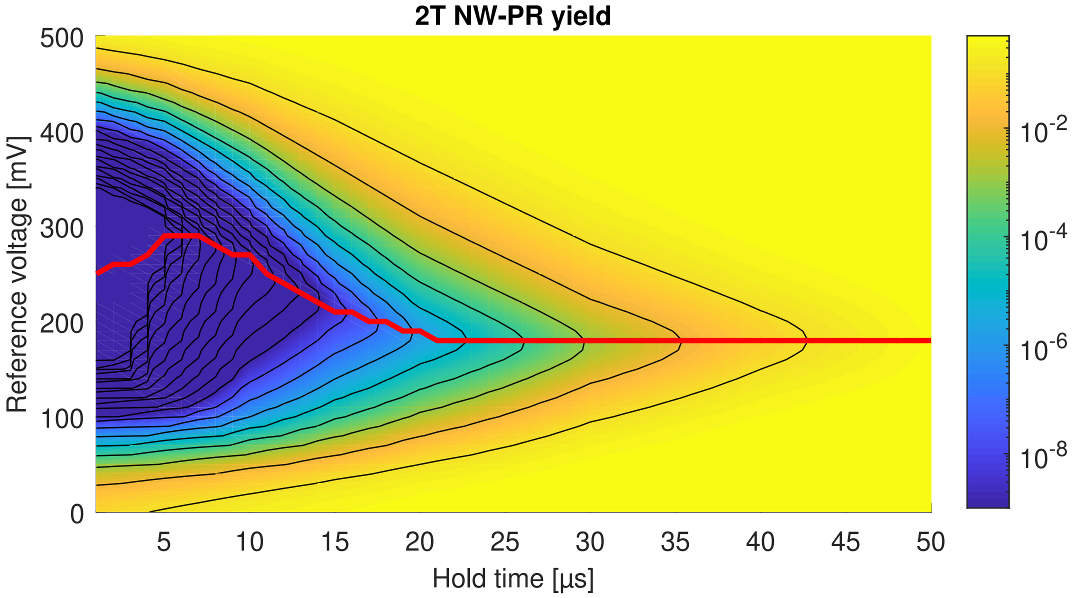


Figure 3.9: 2T NW-PR cell inverse yield for a range of hold times and reference voltages.

input-referred noise standard deviation of $4\,\mathrm{mV}$ are shown in fig. 3.10b. The BER of more than half the cells lies outside the double-precision floating-point range and the probability of a cell having a BER worse than 10^{-11} is roughly $0.0068\,\%$, which corresponds to one expected refresh bit flip every $23.15\,\mathrm{d}$. The median worst cell BER becomes 5.43×10^{-56} and the probability of getting a memory with a worst cell BER above 10^{-11} is roughly $3.3\,\%$.

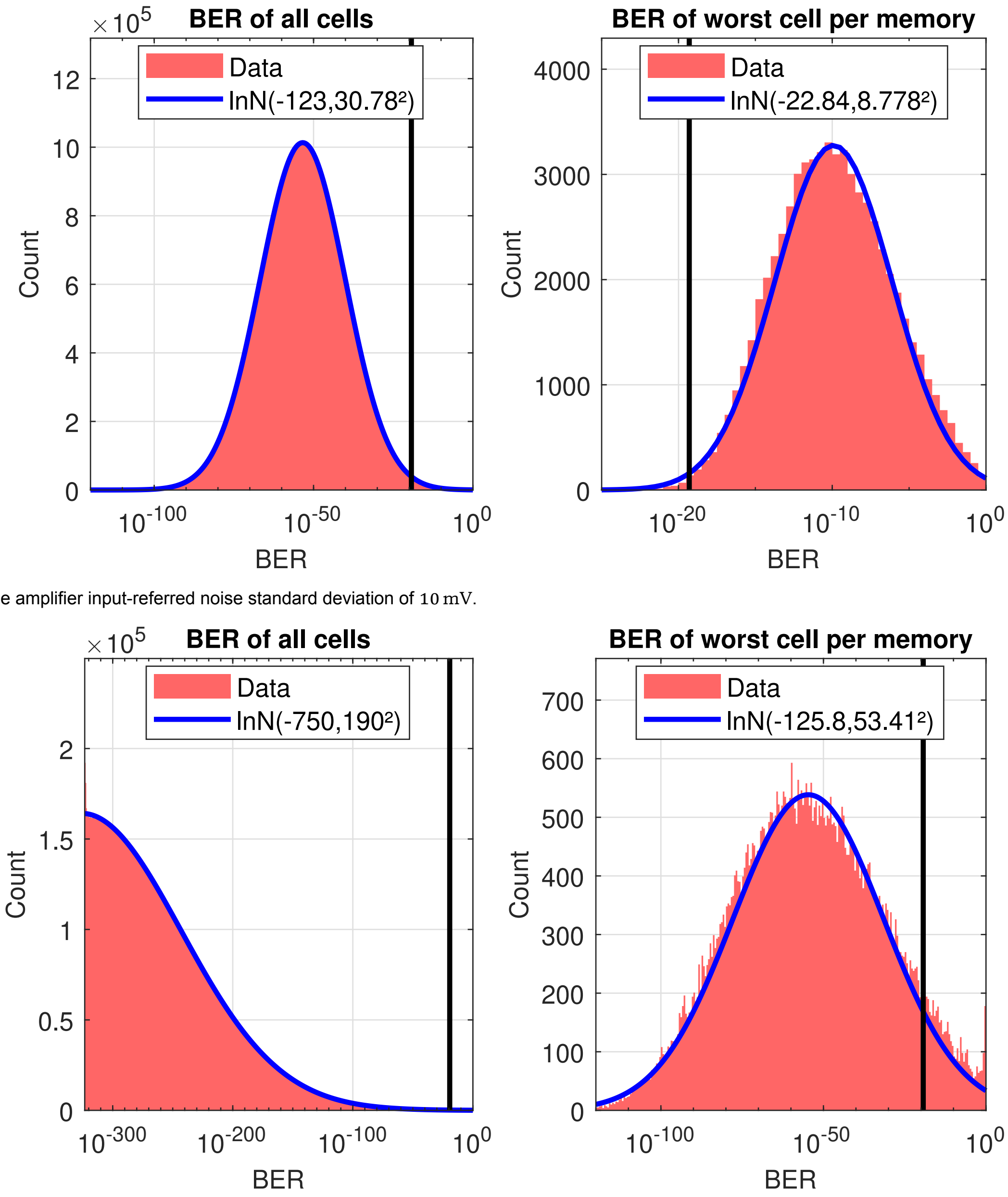
Finally, note that the observed BER for most cells will be much larger than the values calculated using the Monte Carlo simultion. In reality, the BER is not only determined by noise during readout. It is also limited by *soft errors*, for example, due to radiation or interference. Radiation soft errors are reported for both dynamic memories [68], [69] and static memories [70], [71], and put a lower bound on the apparent BER. Using values reported in literature, the expected number of failures in time (FIT), where the time is defined as $10^9\ \mathrm{h}$, for a $1\,\mathrm{kbit}$ array ranges from 0.004 to 10. In $10^9\ \mathrm{h}$, we perform 1.84×10^{20} read operations for a refresh rate of $50\,\mathrm{kHz}$ on 1024 cells. If we assume these failures to be the result of an incorrect refresh read operation, the apparent refresh read error rate will be between 2.17×10^{-23} and 5.43×10^{-20}. This is larger than the noise-based worst cell BER of $93.5\,\%$ of the memories for an input-referred sense amplifier noise standard deviation of $4\,\mathrm{mV}$. The black line in the right graph of figs. 3.10a and 3.10b indicates an apparent BER of 5×10^{-20}.

These BER simulations show that setting the refresh rate for a yield of $1-10^{-6}$ results in good cell performance. Of the just over 67.1 million cells that are simulated, around 180 are either always 0 or 1 without noise. This results in a yield of $\frac{90}{2^{26}}\approx1.34\times10^{-6}$, with the additional factor $\frac{1}{2}$ since these cells will read wrong for only one state. This roughly corresponds with the bad-cell probability of 1.386×10^{-6} at a hold time of $20\,\mathrm{\mu s}$ and reference voltage of $190\,\mathrm{mV}$ from fig. 3.9 using the yield calculation method. Additionally, the observed BER of $99.99\,\%$ of the cells is not limited by noise, but by soft error sources such as radiation. This shows that aiming for a yield of 10^{-6} will result in a much lower BER than 10^{-6} for almost all cells, and good memory performance, where one memory is expected to have a single cell that is not limited by soft errors out of every fifteen memories. Since the performance is good, and to simplify comparison between the different memory designs, the refresh rate of every memory will be chosen such that the yield equals 10^{-6}.

3.2.4. Power

The final memory metric is the power consumption. This will tell us for the given configuration, latency, and desired yield what the most efficient memory type is.

We can split the power consumption into two parts. First, the power that is needed to perform an actual read or write operation, now called the *operational power* will be calculated. Note that this is



(a) Sense amplifier input-referred noise standard deviation of $10\,\mathrm{mV}$.

(b) Sense amplifier input-referred noise standard deviation of $4\,\mathrm{mV}$.

Figure 3.10: Histograms of the BER of 65536 random memories. Left: histogram of the BER of each cell. Right: histogram of the BER of the worst performing cell per memory. The black lines indicate the apparent refresh read BER due to radiation soft errors.

proportional to the read and write frequency and therefore varies over the application space. Second, power is required to keep the data correct, now called the *retention power*. This consists of the cell-leakage power for static cells and refresh power for dynamic cells. This is a constant, minimum amount of power needed by a memory to retain data over the entire application space.

Operational power
The operational power can be split into two parts: the write power and the read power. The power needed for the write operations is the product of the energy needed per write operation and the number of write operations per second. Similarly, the power needed for read operations is the product of the energy needed per read operation and the number of read operations per second.

Write operation energy The write operation energy is determined by all the capacitances that need to be (dis)charged. In this model, the energy needed to (dis)charge and return the following nodes is considered:

- a single wordline (eq. (3.4)),
- half of the bitlines (eq. (3.5)), and
- the gates of bitline precharge transistors (eq. (3.6)).

In these equations, E_{WL}, E_{BL}, and E_{pre} indicate the energy needed to toggle the single wordline, half of the bitline, and the gates of the precharge transistors, respectively. $C_{\mathrm{WWL,cell}}$ and $C_{\mathrm{WBL,cell}}$ are the capacitance added to the write wordline and write bitline by each cell, respectively, and $C_{\mathrm{G,pre}}$ is the effective gate capacitance of a single precharge transistor for full voltage swing. Finally, V_{supply} is the supply voltage, which is always assumed to be $1.1\,\mathrm{V}$. Only half the bitlines are considered since an equal data distribution is assumed. This means that half the bitlines will have to be flipped and half will already be at the correct voltage.

$$E_{\mathrm{WL}} = N_{\mathrm{columns}} \times C_{\mathrm{WWL,cell}} \times V_{\mathrm{supply}}^2 \tag{3.4}$$

$$E_{\mathrm{BL}} = \frac{N_{\mathrm{columns}}}{2} \times N_{\mathrm{rows}} \times C_{\mathrm{WBL,cell}} \times V_{\mathrm{supply}}^2 \tag{3.5}$$

$$E_{\mathrm{pre}} = N_{\mathrm{columns}} \times C_{\mathrm{G,pre}} \times V_{\mathrm{supply}}^2 \tag{3.6}$$

The capacitance values are the sum of the transistor capacitances and the parasitics of a cell connected to each array line. Average transistor capacitance estimates are obtained using simulation models, by averaging the effective capacitance in different operation regimes of the transistors. The effective capacitance is given by the total amount of charge needed to charge a terminal to a certain voltage. The parasitic capacitances are obtained using Parasitic Extraction (PEX) using (preliminary) cell layouts. For the 2T NW-PR cell design and a supply voltage of $1.1\,\mathrm{V}$, this results in the capacitances and energy results shown in table 3.2 where E_{write}, the total energy per write, equals the sum of the aforementioned energy contributions.

Table 3.2: 2T NW-PR write capacitance and energy values.

Capacitance	Value [aF]
$C_{\mathrm{WWL,cell}}$	295.25
$C_{\mathrm{WBL,cell}}$	168.60
$C_{\mathrm{G,pre}}$	160.00

Energy	Value [fJ]
E_{WL}	11.43
E_{BL}	104.45
E_{pre}	6.20
E_{write}	122.08

Read operation energy Similar to the write operation energy, the read operation energy is based on the capacitances that need to be (dis)charged. However, this is not constant since the bitline voltage swing depends on the hold time. In general, the model takes the following nodes into account:

- a single wordline (eq. (3.7)),
- all the bitlines with a 50/50 data distribution (eq. (3.8)),
- enabling and decision energy required for the sense amplifiers (eq. (3.9)), and
- the gates of bitline precharge transistors (eq. (3.10)).

In these equations, E_{WL}, $E_{\text{BL}}(t_{\text{hold}})$, E_{SA}, and E_{pre} indicate the energy required for toggling the single wordline, the bitlines (as a function of the hold time), the sense amplifiers, and the bitline precharge transistors, respectively. $C_{\text{RWL,cell}}$ and $C_{\text{RBL,cell}}$ are the capacitance added to the read wordline and read bitline by each cell, respectively. $C_{\text{SA,in}}$ and $C_{\text{SA,control}}$ are the sense amplifier capacitance at the input and the enabling control node, respectively. $V_{\text{RBL,swing},x}(t_{\text{hold}})$ is the voltage swing on the read bitline for the two data states as a function of the hold time and $E_{\text{SA,decision}}$ is the energy required by the sense amplifier to make a decision. Contrary to the write operation energy, we need to consider all the bitlines. Since both states develop a bitline swing, albeit of different magnitude, we need to include all of them.

$$E_{\text{WL}} = N_{\text{columns}} \times C_{\text{RWL,cell}} \times V_{\text{supply}}^2 \tag{3.7}$$

$$E_{\text{BL}}(t_{\text{hold}}) = N_{\text{columns}} \times (N_{\text{rows}} \times C_{\text{RBL,cell}} + C_{\text{SA,in}}) \times V_{\text{supply}} \times \frac{V_{\text{RBL,swing,0}}(t_{\text{hold}}) + V_{\text{RBL,swing,1}}(t_{\text{hold}})}{2} \tag{3.8}$$

$$E_{\text{SA}} = N_{\text{columns}} \times \left(E_{\text{SA,decision}} + C_{\text{SA,control}} \times V_{\text{supply}}^2\right) \tag{3.9}$$

$$E_{\text{pre}} = N_{\text{columns}} \times C_{\text{G,pre}} \times V_{\text{supply}}^2 \tag{3.10}$$

Additionally, in the case of the 2T NW-PR cell design, we have to take the readout leakage energy into account. The readout leakage current is caused by all cells with state 0 in the same column as the cell with state 0 that is being read. If the read bitline is charged to a sufficiently large voltage, the read transistors of the unselected cells are put in inversion and leak away some of the readout current. Since this only happens for large bitline voltage, it will increase the read energy for short hold times, where the 0 state is still very strong and leads to fast bitline charging. The total energy lost due to this leakage for a read duration of $1\,\text{ns}$ can be seen in fig. 3.11 in red against the hold time and is determined based on a simulation of a single column with all cells in state 0. This leakage is halved to assume only half of the unselected cells are in state 0 and multiplied by half the number of columns to assume that half the read cells are in state 0.

The read energy is now computed similar to the write energy. The combined transistor and parasitic capacitances are shown in table 3.3. The general read energy contributions can then be calculated and added together with the readout leakage energy to give the total read energy. Figure 3.11 also shows the total energy required for a read operation after a specific hold time. Assuming that read operations are spread uniformly between a zero hold time and the refresh period ($19.64\,\mu\text{s}$), the average read energy is equal to $183.03\,\text{fJ}$ per operation.

Table 3.3: 2T NW-PR read capacitance values.

Capacitance	Value [aF]
$C_{\text{RWL,cell}}$	372.37
$C_{\text{RBL,cell}}$	284.88
$C_{\text{SA,in}}$	720.00
$C_{\text{SA,control}}$	394.00

Retention power

Finally, the retention power can be computed. For static cells, this consists of cell leakage currents and the total retention power can be calculated using eq. (3.11) where $I_{\text{leak,cell}}$ is the leakage current of an individual cell. For dynamic cells, we need to perform a read and a write on all rows at the refresh rate which leads to the power consumption shown in eq. (3.12). This is significantly larger than the leakage

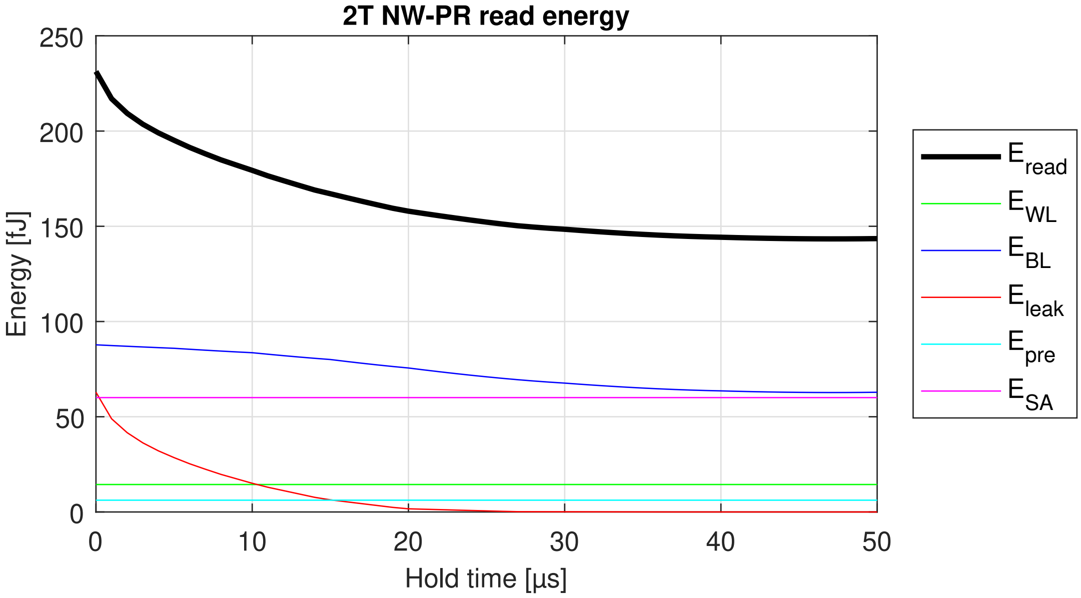


Figure 3.11: 2T NW-PR read energy and its components as a function of the hold time.

that causes the storage-node voltage to drift slightly, since the storage-node capacitance is small. The storage-node leakage is therefore ignored. Note that the read energy is taken for a hold time equal to the retention time and is therefore slightly smaller than the average read energy due to reduced bitline swing. Leakage from the peripherals is ignored, since it is expected to be small, especially with a large cell array.

$$P_{\text{ret,static}} = N_{\text{cells}} \times I_{\text{leak,cell}} \times V_{\text{supply}} \tag{3.11}$$

$$P_{\text{ret,dynamic}} = N_{\text{rows}} \times (E_{\text{read}}(t_{\text{hold}} = t_{\text{retention}}) + E_{\text{write}}) \times f_{\text{refresh}} \tag{3.12}$$

For the 2T NW-PR cell design, this results in a combination of previously computed values. With 32 rows, a refresh rate of $50.91\,\text{kHz}$, a write energy of $122.08\,\text{fJ}$, and a read energy at a hold time of $19.64\,\mu\text{s}$ of $158.48\,\text{fJ}$, we obtain a total retention power of $457.05\,\text{nW}$.

3.2.5. Trade-offs

As mentioned before, there are trade-offs between the different metrics. We will briefly look at two possible changes and their effects, namely changing the read operation duration and the target yield.

To improve the read latency, the duration of a read operation can be decreased. This will cause the bitline voltages to lie closer together, resulting in a worse yield/BER for the same refresh rate or a higher refresh rate and therefore higher retention power for the same BER. However, the reduced bitline swing will also reduce the read operation energy, which could improve the total power consumption if the average read frequency is high.

In some applications, a higher BER may be allowed due to, for example, the use of error correcting codes (ECC). This would allow for a reduction of the refresh rate and therefore the retention power at the cost of the energy for the error decoding hardware. This could improve the total power consumption if the average read frequency is low and the additional area consumption and complexity is allowed.

Due to these trade-offs, a selection of parameters has to be made based on the application. As mentioned before, in this model we will assume the same yield target for each of the memories of $1 - 10^{-6}$. This allows for a better comparison of the memory cells. In the next section, we will see how the memory operation duration needs to be changed for the alternative memory cell designs and what changes in the calculation of the metrics, taking cell-specific design aspects into account.

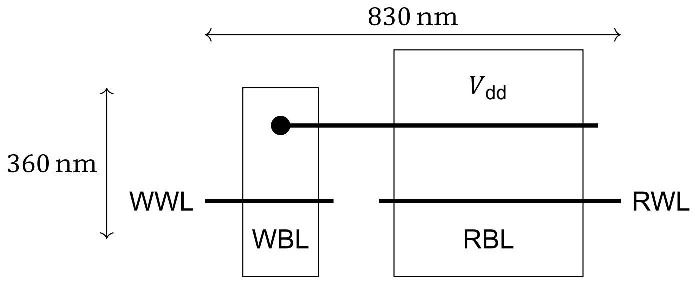


Figure 3.12: Simplified layout of a 3T NW-PR cell design with the write transistor (NW) on the left side and the readout transistors (PR and PR') stacked vertically on the right.

3.3. Memory cells

In the previous section, it has been shown how the model can be used to calculate the metrics and some of the degrees of freedom in memory array design. This was done with the 2T NW-PR cell design as an example. In this section, the model is extended to include the other memory cell designs. The differences between the cell designs are briefly repeated, followed by how this translates into the memory model, and the effects it has on the memory metrics. In the end, the results of all the cells are compared and a method of picking the best memory cell design for each application is shown.

3.3.1. Dynamic 3T NW-PR

The dynamic 3T NW-PR cell is a derivative of the 2T NW-PR cell with an additional readout transistor. This significantly reduces the coupling between the read wordline and the storage node. Additionally, this removes the readout leakage problem of the 2T cell. A cell schematic and further operational details can be found in section 2.1.2. In the following sections, we will see the effects this design changes has on the memory metrics.

Area

The area increases due to the need for an additional transistor. The peripherals can mostly be the same, so we will assume that their dimensions stay constant. The cell height remains the same, while the width increases slightly since the readout transistor width of neighbouring cells can not be shared, as shown in the simplified layout in fig. 3.12. This results in the dimension shown in table 3.4, a cell area of $0.2988\,\mathrm{\mu m}^2$, and a total area of $584\,\mathrm{\mu m}^2$ (for $N_{\mathrm{columns}} = 32$ and $N_{\mathrm{rows}} = 32$).

Table 3.4: 3T NW-PR array dimension estimations

Dimension	w_{cell}	h_{cell}	w_{peri}	h_{peri}
$\mathrm{\mu m}$	0.83	0.36	8	7

Latency

During a write operation, this cell type is as fast as the 2T NW-PR cell. It also writes to a PMOS gate through an NMOS pass transistor in exactly the same way. Again, the write duration is set as long as the read duration to maximise the storage-node voltage difference, while not limiting the combined operation latency.

The additional readout transistor reduces the coupling between the read wordline and the storage node. This results in a larger overdrive on the read transistor and therefore a stronger cell and thus a faster read operation. This can be seen in fig. 3.13, with read times of $100\,\mathrm{ps}$, $250\,\mathrm{ps}$, $400\,\mathrm{ps}$, $550\,\mathrm{ps}$, $700\,\mathrm{ps}$, $850\,\mathrm{ps}$ and $1000\,\mathrm{ps}$, where the bitlines can get charged close to or over $1\,\mathrm{V}$.

The obtained bitline margin for a given read duration for various hold times is shown in fig. 3.14. Although the optimum again lies at roughly the same read duration as for the 2T NW-PR cell (around $1\,\mathrm{ns}$ for a hold time of $10\,\mathrm{\mu s}$), a lower value is chosen to limit the total bitline swing and therefore the read operation energy consumption. To obtain similar bitline voltage levels as the 2T NW-PR cell, we reduce the read latency to $250\,\mathrm{ps}$.

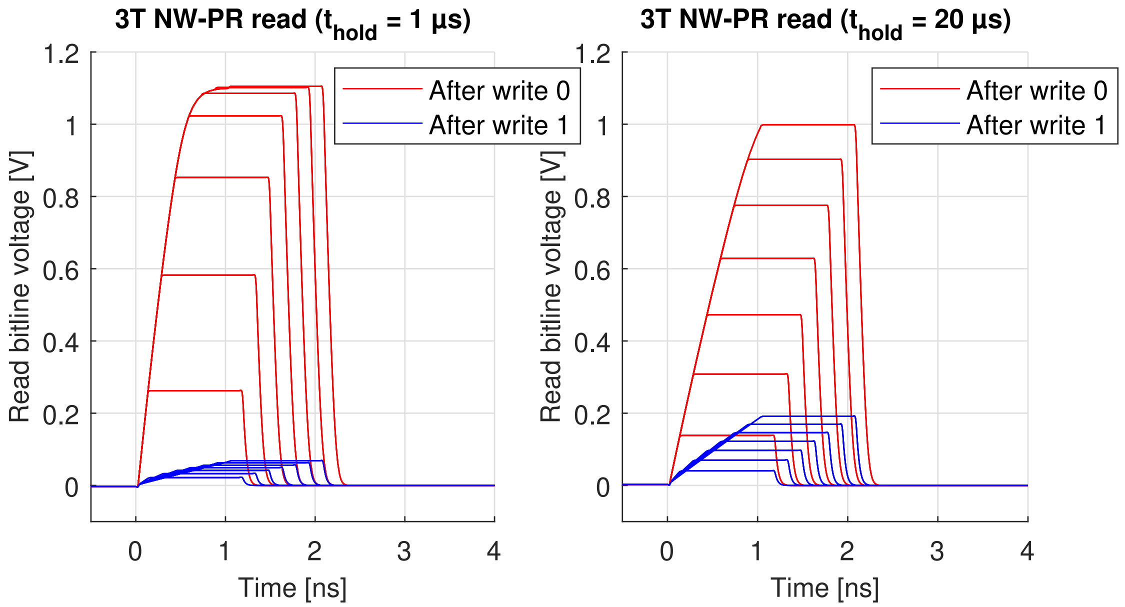


Figure 3.13: 3T NW-PR bitlines during a read operation with durations of $100\,\mathrm{ps}$, $250\,\mathrm{ps}$, $400\,\mathrm{ps}$, $550\,\mathrm{ps}$, $700\,\mathrm{ps}$, $850\,\mathrm{ps}$ and $1000\,\mathrm{ps}$. Left: read $1\,\mu\mathrm{s}$ after write, right: read $20\,\mu\mathrm{s}$ after write.

Yield

The leakage of this cell from the storage node at the write transistor (NW) side will be similar to the 2T NW-PR cell. However, the readout transistor (PR) always has the supply voltage connected to one of the terminals. Especially state 0 will therefore experience a leakage current pulling the storage-node voltage up. This causes the readout current for the low state to drop faster, resulting in a lower retention time. This means a lower yield for the same refresh rate or a higher refresh rate for the same yield, when compared with the 2T NW-PR cell design. This can be seen in the graph shown in fig. 3.15. Finding the optimal point for a yield of $1-10^{-6}$ results in an optimal reference of $164.4\,\mathrm{mV}$, a retention time of $17.12\,\mu\mathrm{s}$, and a refresh rate of $58.42\,\mathrm{kHz}$.

Notice that the reference voltage is lower than for the 2T NW-PR cell while the read bitline voltage is typically larger. As we can see in fig. 3.13, the bitlines are also charged slightly if the cell has been written high. Since the high write is weak and short and there is no coupling step like in the 2T NW-PR cell design, the readout transistor is in weak inversion even for a high storage node voltage. Intuitively, this means that the optimal reference must lie higher than for the 2T NW-PR cells. However, as mentioned before, the 3T NW-PR cell mainly suffers from leakage pulling state 0 up, while the 2T NW-PR cell mainly suffers from leakage pulling state 1 down. This causes the two states to meet at a lower read bitline voltage, and therefore requires a lower reference voltage.

Power

The increased cell width results in a larger wordline capacitance since the lines get longer. This will increase the wordline energy for both operations. Additionally, the write bitline is enclosed by more metal on the second metal layer, resulting in an increased coupling capacitance. However, due to the removal of readout leakage, a net reduction of energy per read operation is achieved. Additionally, the capacitive load on the read bitline has decreased, which reduces the bitline energy for a read operation. The larger cell size allows for larger metal spacing, which reduces the coupling between lines. Due to the increased refresh rate for similar yield, the retention power is higher than that of the 2T NW-PR cell.

The cell capacitance per array line is shown in table 3.5. The sense amplifier and precharge capacitances are the same as for the 2T NW-PR cell. The wordline capacitances and write bitline capacitance increased, while the read bitline capacitance has decreased. The wordline layout is similar to the wordlines in the 2T NW-PR cell, so the increase in cell area causes their total capacitance to increase slightly. The read bitlines are further from other metals, especially on the first metal layer, which reduces their parasitic capacitance significantly ($-25\,\%$).

The resulting operational energies and retention power can be seen in table 3.6. Compared to the

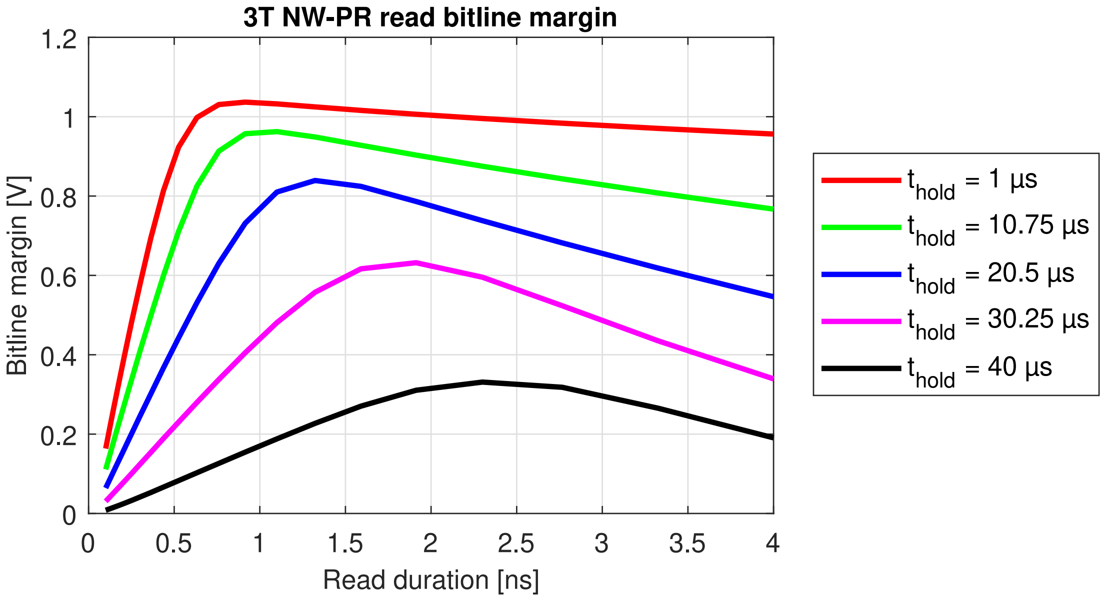


Figure 3.14: 3T NW-PR read bitline margin as a function of the read duration for various hold times.

Table 3.5: 3T NW-PR cell and precharge transistor capacitance values.

Capacitance	$C_{\text{WWL,cell}}$	$C_{\text{WBL,cell}}$	$C_{\text{RWL,cell}}$	$C_{\text{RBL,cell}}$
Value $[\text{aF}]$	332.40	216.57	385.05	213.34

2T NW-PR, the total write energy is higher, but the average read energy is lower. The total energy needed to refresh a single row is higher, and the reduction in retention time to $17.12\,\mu\text{s}$ results in a refresh rate of $58.42\,\text{kHz}$ and a total refresh power for 32 rows of $541.99\,\text{nW}$. This is almost $85\,\text{nW}$ higher than for the 2T NW-PR cell, or an increase of over $18.5\,\%$.

Table 3.6: 3T NW-PR write, read, and refresh energies.

Energy	E_{WL}	E_{BL}	E_{pre}	E_{write}	E_{WL}	$\overline{E_{\text{BL}}}$	E_{SA}	$\overline{E_{\text{read}}}$	$E_{\text{read}}(t_{\text{retention}})$	E_{refresh}
Value $[\text{fJ}]$	12.87	134.17	6.20	153.24	14.91	69.82	60.06	150.98	136.69	289.92

3.3.2. Dynamic 3T PW-PR

The preferential boosted, dynamic 3T PW-PR cell [20] is similar to the 3T NW-PR cell, but uses readout coupling to improve the difference between storage-node voltages for the different states. It also does not have the readout leakage problem of the 2T cell. A cell schematic and further operational details can be found in section 2.1.2. Next, the changes in the memory metrics due to this design change are covered.

Area

The area is similar to the 3T NW-PR since we again need three transistors in a similar arrangement. Since all transistors are P-type, we may expect to be able to place them closer together. However, since the read wordline is now also needed at the source of the readout transistor instead of it being a supply node which can be shared between neighbouring cells, slightly more space is needed (cell design details can be found in chapter 4). A simplified layout is shown in fig. 3.16, which results in the dimension shown in table 3.7, a cell area of $0.3132\,\mu\text{m}^2$, and a total area of $608\,\mu\text{m}^2$ (for $N_{\text{columns}} = 32$ and $N_{\text{rows}} = 32$). The peripheral dimensions are assumed to stay constant.

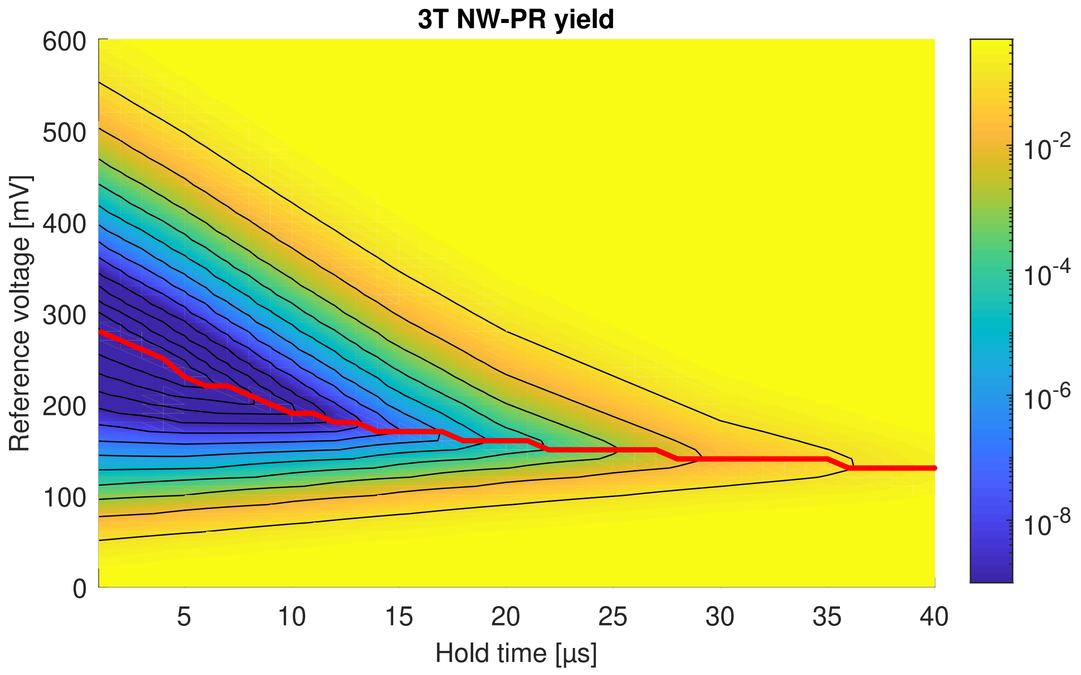


Figure 3.15: 3T NW-PR cell inverse yield for a range of hold times and reference voltages.

Table 3.7: 3T PW-PR array dimension estimations

Dimension	w_{cell}	h_{cell}	w_{peri}	h_{peri}
µm	0.87	0.36	8	7

Latency

In this memory cell design, we write through a P-type pass transistor. This means that now the high state will be written with ease, while the low state voltage will be increased with approximately one threshold voltage. Since hole mobility is lower than electron mobility, the write speed will be slightly lower. Again, we will set the write latency equal to the read latency to ensure that the write operation develops the maximum storage-node voltage difference without limiting the combined operation latency.

Unlike the NW-PR variant of the 3T cell, this cell has a slow read operation similar to the 2T NW-PR cell. During a read operation, the bitline is discharged through a PMOS stack with limited overdrive, which limits the current that can be drawn. The bitline voltages during a read operation can be seen in fig. 3.17 for read durations of 100 ps, 250 ps, 400 ps, 550 ps, 700 ps, 850 ps and 1000 ps. The read duration is fixed to 1 ns like for the 2T NW-PR cell, since it still gives a reasonable bitline swing without being too slow.

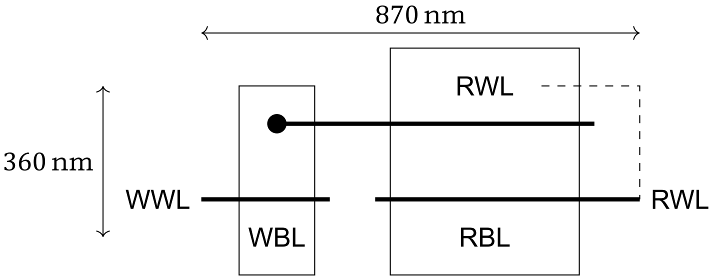


Figure 3.16: Simplified layout of a 3T PW-PR cell design with the write transistor (PW) on the left side and the readout transistors (PR and PR') stacked vertically on the right.

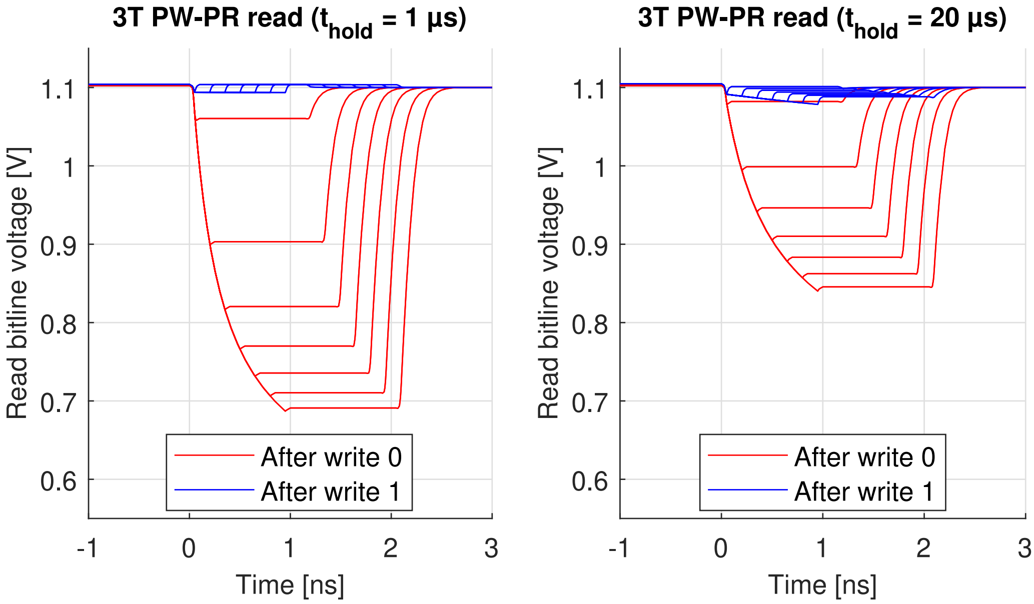


Figure 3.17: 3T PW-PR bitlines during a read operation with durations of $100\,\mathrm{ps}$, $250\,\mathrm{ps}$, $400\,\mathrm{ps}$, $550\,\mathrm{ps}$, $700\,\mathrm{ps}$, $850\,\mathrm{ps}$ and $1000\,\mathrm{ps}$. Left: read $1\,\mathrm{\mu s}$ after write, right: read $20\,\mathrm{\mu s}$ after write.

Yield

The leakage from this cell is similar to that of the 3T NW-PR cell, except there is a P-type write pass transistor instead of N-type. Although the difference between the bitline voltages (bitline margin) starts off smaller than that of the 3T NW-PR cell design, due to the weaker readout since the read bitline is discharged through a PMOS transistor, the bitline margin collapses slower. This means that the yield will be slightly better than that of the 3T NW-PR cell design. This can be seen in the yield graph in fig. 3.18. Again, for a desired yield of $1 - 10^{-6}$, we find an optimal reference of $975\,\mathrm{mV}$, a resulting retention time of $18.11\,\mathrm{\mu s}$, and therefore a refresh rate of $55.23\,\mathrm{kHz}$.

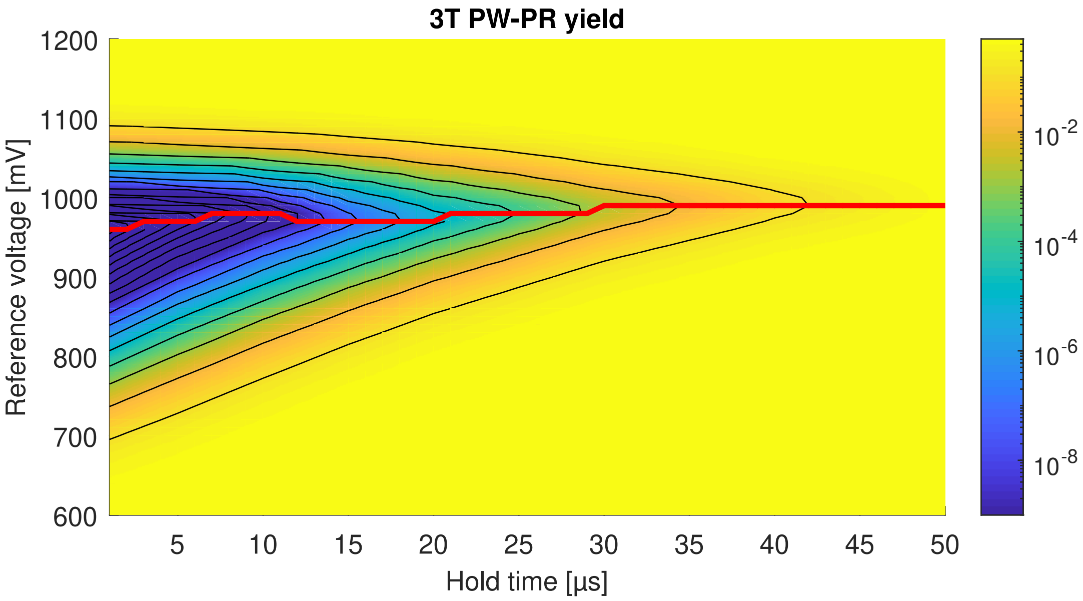


Figure 3.18: 3T PW-PR cell inverse yield for a range of hold times and reference voltages.

Power
Due to the connection of the read wordline to the source of the readout transistor, the routing is more complex and the cell is slightly larger. This leads to a higher capacitance of the array lines when compared to the other dynamic cells, which increases the energy consumption per operation. However, the swing on the read bitlines is much smaller, reducing the read energy below that of the 2T NW-PR cell.

The capacitances of the array lines per cell are shown in table 3.8. The capacitances of the sense amplifiers and precharge transistors are assumed to be the same as for the other cells.

Table 3.8: 3T PW-PR cell capacitance values.

Capacitance	$C_{\text{WWL,cell}}$	$C_{\text{WBL,cell}}$	$C_{\text{RWL,cell}}$	$C_{\text{RBL,cell}}$
Value $[\text{aF}]$	339.74	286.73	599.93	308.48

Table 3.9 shows the write, read, and refresh energies for the 3T PW-PR cell design. At a refresh rate of $55.23\,\text{kHz}$ and 32 rows, the total retention power equals $605.45\,\text{nW}$.

Table 3.9: 3T PW-PR write, read, and refresh energies.

Energy	E_{WL}	E_{BL}	E_{pre}	E_{write}	E_{WL}	$\overline{E_{\text{BL}}}$	E_{SA}	$\overline{E_{\text{read}}}$	$E_{\text{read}}(t_{\text{retention}})$	E_{refresh}
Value $[\text{fJ}]$	13.15	177.63	6.20	196.98	23.23	68.49	60.06	157.97	145.57	342.56

3.3.3. Static 6T

Finally, the static 6T cell is the most used static memory cell design. It does not require refreshing operations and is used slightly different than the dynamic cells, however most of the calculation methods for memory metrics of the dynamic cells are still valid. A cell schematic and further operational details can be found in section 2.1.2.

Area
The 6T static cell is much larger than the dynamic cells, since it requires six transistors instead of three. A preliminary layout is created using the *Lithographically Symmetrical* (LS) cell layout which has been common since $90\,\text{nm}$ technology nodes [21] and is shown in fig. 3.19. The cell dimensions are shown in table 3.10. The resulting memory cell has an area of $0.5376\,\mu\text{m}^2$.

The peripheral dimensions are also shown in table 3.10. The width of the row decoder is assumed to roughly halve, since the 6T static cell only has one set of wordlines instead of two separate lines for reading and writing, reducing the total number of wordlines from 64 to 32. This means that we will use only one decoder for reading and writing instead of two decoders for reading and writing separately. This saves some area and results in a total array area of $822\,\mu\text{m}^2$ (for $N_{\text{columns}} = 32$ and $N_{\text{rows}} = 32$).

Table 3.10: 6T array dimension estimations

Dimension	w_{cell}	h_{cell}	w_{peri}	h_{peri}
μm	1.28	0.42	4	7

Latency
For a static cell, the stored data does not improve with the duration of a write operation. As shown in the left graph of fig. 3.20, the feedback takes over the data applied for the write within $50\,\text{ps}$ to $100\,\text{ps}$. This means that the write latency for these static 6T cells can be very low, again resulting in a combined operation latency limited by the read operation latency.

The read operation on the 6T static cell is fast, similar to the 3T NW-PR cell. The right figure of fig. 3.20 shows the two bitlines during read operations with durations of $100\,\text{ps}$, $150\,\text{ps}$, $200\,\text{ps}$, $250\,\text{ps}$, $300\,\text{ps}$, $350\,\text{ps}$, $400\,\text{ps}$, $450\,\text{ps}$ and $500\,\text{ps}$. The low side bitline is pulled down through an NMOS stack, which is very fast. Since the bitline levels also do not collapse over time, we pick a read duration of $150\,\text{ps}$. This results in a low side bitline level of roughly $650\,\text{mV}$, or a bitline margin of $450\,\text{mV}$.

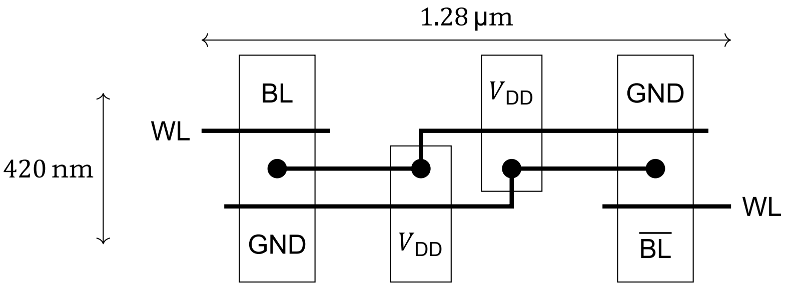


Figure 3.19: Simplified LS layout of the 6T cell design with the NMOS transistors (ND/NA and ND'/NA') on either side and the PMOS transistors (PL and PL') in the middle.

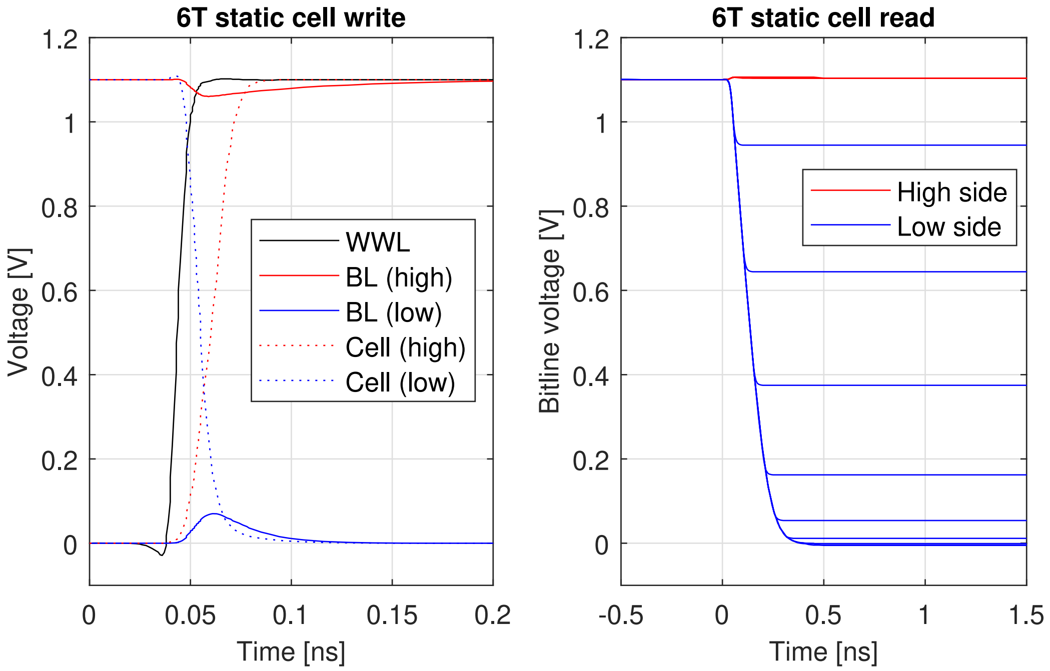


Figure 3.20: Left: 6T static cell internal nodes during write operation. Right: 6T static cell bitlines during read operations with durations of 100 ps, 150 ps, 200 ps, 250 ps, 300 ps, 350 ps, 400 ps, 450 ps and 500 ps.

Yield

The yield of the static cells is much better than that of the dynamic cells. Contrary to the dynamic cells, the bitline margin for static cells does not shrink depending on the hold time. Additionally, the sense amplifier only compares the two bitlines and there is no reference voltage involved. From a Monte Carlo simulation of the cells, we find that the bitline margin follows a normal distribution with mean $458.6\,\mathrm{mV}$ and a standard deviation of $50.84\,\mathrm{mV}$.

The yield can be computed by comparing the bitline margin distribution with the offset distribution of the sense amplifiers. An error occurs when the bitline margin is smaller than the offset (ignoring metastability overhead), which is shown in eq. (3.13), where M is the bitline margin distribution and O is the sense amplifier offset distribution. Instead of comparing the distributions, their difference C can be compared with zero. The difference can be computed by taking the difference of the means and adding the variances. The probability of the comparison distribution C being smaller than zero, and therefore the yield, equals 4.75×10^{-18}.

$$\begin{aligned} &P(M < O), \text{ where } M \sim N(0.4586, 0.05084^2) \text{ and } O \sim N(0, 0.0165^2) \\ &C = M - O, \text{ where } C \sim N(0.4586, 0.05084^2 + 0.0165^2) \\ &P(M < O) = P(C < 0) \approx 4.75 \times 10^{-18} \end{aligned} \tag{3.13}$$

Power

The energy and power values for the static cell are computed slightly different than those for the dynamic cells. The first major difference for the read and write operation energies is that the bitlines work complementary. This means that one of the bitlines will always be discharged, regardless of the data. This effectively doubles the bitline capacitance, giving it the highest total bitline capacitance of all cell designs. Second, when the cell flips during a write operation, it uses a fixed amount of charge. This is captured in a constant amount of energy that is added to the write operation energy. Finally, the static cell does not require refreshes to maintain its data, but instead has a small leakage current through its inverter pair. This leakage current determines the retention power of the static cell.

The capacitance values for the static cell are shown in table 3.11. Note that this does not include the virtual capacitance doubling due to the differential bitline operation. Additionally, since the bitline voltages are different during read and write operations, the effective capacitances seen from the wordlines is slightly different for the two operations.

Table 3.11: 6T static cell capacitance values.

Capacitance	$C_{\text{WWL,cell}}$	$C_{\text{RWL,cell}}$	$C_{\text{BL,cell}}$
Value $[\mathrm{aF}]$	498.67	485.94	187.88

Using the capacitance values, we can calculate the energy and power metrics. The energy required for the write and read operations is shown in table 3.12. The part of the write energy denoted as E_{cells} indicates the energy needed by half the cells to flip. The leakage energy can be computed from the leakage current of each cell which equals $20.47\,\mathrm{pA}$ at a supply of $1.1\,\mathrm{V}$. For 1024 cells, this results in a total leakage power of $23.06\,\mathrm{nW}$.

Table 3.12: 6T static cell write and read energies.

Energy	E_{WL}	E_{BL}	E_{pre}	E_{cells}	E_{write}	E_{WL}	E_{BL}	E_{SA}	E_{read}
Value $[\mathrm{fJ}]$	19.31	260.67	6.20	17.71	303.88	18.82	106.64	60.06	191.70

3.3.4. Comparison

Using the metrics computed for the various cells, we can compare the different designs. The final metrics are shown in table 3.13. We can see by the colours that each design has some well and poor performing metrics. For example, the 2T NW-PR cell is the smallest in terms of area, has the lowest write energy of all cell designs, and has the lowest retention power of the dynamic cell designs. However, it also has the highest read energy of the dynamic cell designs. It is therefore a good cell choice for large and high activity memories with a lot of write operations. However, some applications may be heavily read operation dominated. If the memory is not restricted by area, one of the other dynamic cell designs or the static cell design may be better suited.

Table 3.13: Comparison of memory metrics for four different cell designs.

Metric	2T NW-PR	3T NW-PR	3T PW-PR	6T
Area $[\mu\mathrm{m}^2]$	466	584	608	822
Latency $[\mathrm{ns}]$	1	0.25	1	0.15
Cell yield	10^{-6}	10^{-6}	10^{-6}	4.75×10^{-18}
Write energy $[\mathrm{fJ}]$	122.08	153.24	196.98	303.88
Read energy $[\mathrm{fJ}]$	183.03	150.98	157.97	191.70
Retention power $[\mathrm{nW}]$	457.05	541.99	605.45	23.06

To show the trade-offs over the application space, the write and read energy are combined with the retention power to create a *memory landscape*. The total power consumption over the application space is computed for each memory, and the memory with the lowest total power consumption is plotted. This memory landscape is shown in fig. 3.21.

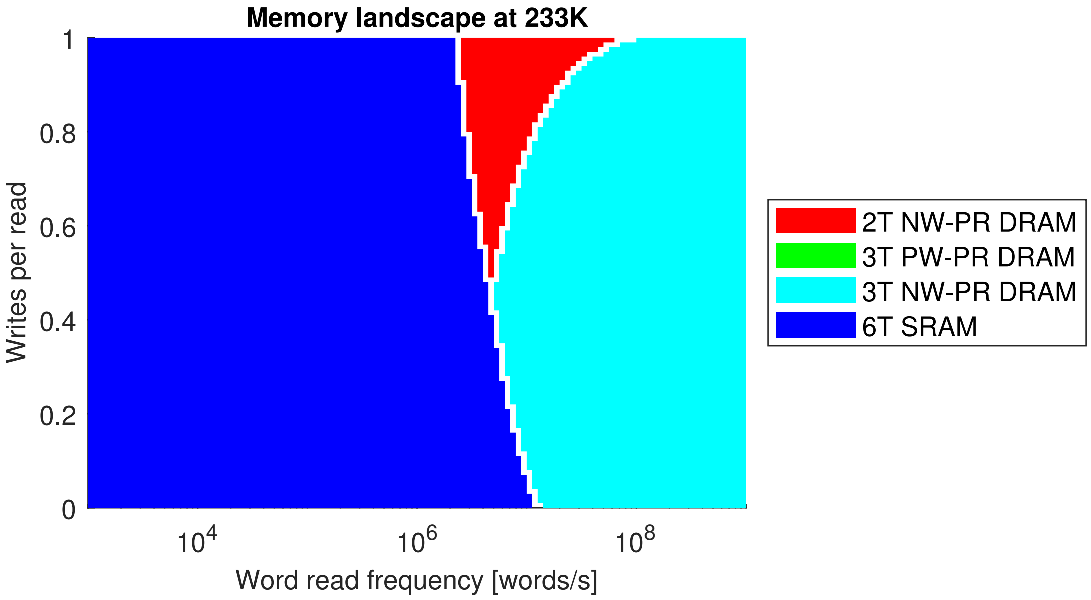


Figure 3.21: Memory landscape showing the memory with the lowest power consumption over the application space with read frequencies from 1 kHz to 1 GHz and average number of writes per read between 0 and 1.

The memory landscape shows a clear separation between the static and dynamic cell designs. For low frequency operation (less than 2×10^6 words read per second), the static 6T cell design beats the dynamic cells due to its low retention power. As the read frequency increases, the dynamic cells start outperforming the static cell design. At high operation frequencies, the write and read power dominate the total power consumption, and the 6T cell design has the highest write and read energies of all cell designs.

In the dynamic cell region of the memory landscape, there are two regions which depend on the write to read operation ratio. In a read only application, the cell design with the lowest read energy will give the lowest power consumption, which is the 3T NW-PR cell design. In applications where there are as many writes as reads, such as queues, the cell design with the lowest added read and write energy will give the lowest power consumption. This is also the 3T NW-PR cell design, but for read frequencies around 10^7 words per second, the low retention power of the 2T NW-PR cell design offsets the additional operation power. Since the 3T NW-PR cell design outperforms the 3T PW-PR cell design in every metric, the 3T PW-PR cell is not seen in the landscape as its power consumption is never the lowest.

3.4. Moving to 4.2 K

In this section, we will see how we can model cryo-CMOS effects by changing various aspects of the model. First, we will see how changes due to cryo-CMOS affect the metrics. This is followed by an overview of the modelled changes. Finally, we will see how these affect the memory landscape over the application space.

3.4.1. Cryo-CMOS

When cooling CMOS devices to cryogenic temperatures, specifically to 4.2 K, their device characteristics change. We will first list some of the main effects that can be expected and the effect they have on the memory metrics. Next, we will list how some of these effects are modelled.

Increased subthreshold slope The increase in subthreshold slope has a huge effect on the dynamic cells as it severely reduces the subthreshold leakage through the write transistors. This means that the retention time of the dynamic cells is expected to increase significantly. This results in a reduced refresh rate and retention power consumption.

The same holds for the static cells. A decrease in leakage current will reduce the current flowing through the two inverters. This will lower the retention power consumption of the entire static array.

Increased threshold The increase in threshold voltage will increase the read latency slightly for obtaining the same bitline voltage or lower the yield/BER for the same read latency. Because the overdrive on the readout transistors will be lower, the bitline (dis)charging currents will be smaller. This results in smaller bitline margins and, therefore, increases the probability of offset and noise causing an incorrect comparison.

Additionally, the retention time will be reduced slightly for dynamic cells. Since the write voltage passes through a single pass transistor, one of the voltage levels will be reduced by the threshold voltage. Increase in the threshold voltage therefore leads to a reduced difference between the two storage-node voltage levels. This will lead to faster collapsing bitline voltages and therefore reduce the retention time.

Finally, the gate capacitance will decrease slightly. Due to the higher threshold, less charge is needed to charge the gate to the supply voltage. This means that the effective capacitance of the wordlines will decrease slightly, lowering the energy needed for a read or write operation.

Reduced active region capacitance and leakage The reduction of the capacitance of the source-to-bulk and drain-to-bulk junction diodes will lower the capacitance attached to wordlines and bitlines, resulting in lowered read and write operation energy. Additionally, the leakage through the source-to-bulk and drain-to-bulk diodes decreases. This leads to a higher retention time for dynamic cells, and less leakage for static cells. This results in a lower retention power consumption for both cell types, similar to the decreased subthreshold leakage.

Increased mobility The carrier mobility in devices increases, which leads to a larger current for the same overdrive voltage. This will increase the speed of readout by increasing the current of the readout transistor. However, this effect is reduced, or can even be cancelled, by the increase in threshold voltage.

Increased mismatch The mismatch between devices increases at cryogenic temperatures. This means that the variation between cells and sense amplifiers is expected to increase, resulting in lower yield for the same refresh period. To keep a constant yield, this means that we will need to increase the refresh rate of dynamic cells and therefore increase the retention power.

Reduced interconnect resistance The reduced resistance of metal interconnect at cryogenic temperatures causes the RC time constant of wires to reduce. This results in slightly faster signal propagation and could therefore also speed up the read and write operations, resulting in lower latency. However, this will only be noticeable for long lines, where the propagation delay due to the RC time constant of the line plays a significant role in the total operation latency.

Reduced noise The reduced temperature causes noise to decrease significantly. As a result, the BER distributions (such as those shown in fig. 3.10) will move even further to the left, resulting in much better average cell performance. However, since the yield decreases due to increased mismatch, the right tail of the distribution will be slightly larger.

Modelled effects

Only the most relevant cryo-CMOS effects have been modelled, neglecting aspects that were too complex for simple analytical model, aspects that roughly cancel out, or that are expected to be insignificant. We will list three effects that have been modelled and explain how they are modelled.

Reduced leakage The reduction in leakage due to reduced subthreshold and diode leakage is modelled for both dynamic and static cells. For the dynamic cells, the leakage reduction is modelled by multiplying the x-axis of the simulated bitline voltage curves (such as those shown in fig. 3.7) by a factor $50\,000$. This results in expected retention time of about $1\,\mathrm{s}$ instead of $20\,\mu\mathrm{s}$, which is in line with expectation from literature [56]. The leakage of static cells is assumed to go to zero. This is optimistic as there will still be some leakage, but it is hard to predict accurately and expected to be negligible compared to the dynamic cell retention power.

In reality, the amount of leakage reduction depends on the activation energy of the dominant leakage source. This causes the subthreshold leakage of the write pass transistors to drop significantly, until it is no longer dominant. Leakage sources based on tunnelling effects, such as GIDL at the write transistors and gate leakage at the readout transistors, have a much lower activation energy. This means that they are less temperature dependent and are therefore expected to limit the retention time [13].

The readout leakage energy from the 2T NW-PR cell has also been reduced. Due to the increase in threshold voltage, the total leakage energy is assumed to halve.

Reduced capacitance The reduction of capacitance for the transistor terminals (gate, source, and drain) to the substrate is modelled by scaling the effective capacitances with a fixed factor. The wordline and bitline capacitances are computed as the sum of metal (parasitic) capacitances and transistor capacitances. Some of the parasitic capacitances are between metal and the substrate and their effect may be reduced to the increased substrate impedance in series with the capacitance. However, most parasitic capacitances are between metals and do not scale. Therefore, the parasitic capacitances are assumed to remain constant and only the transistor capacitances are scaled. The scaling factor is assumed to be 0.8× [37], but the memory landscape is not sensitive to the exact value.

Increased mismatch Finally, the increase in mismatch is also partially included in the 4.2 K model. Changes in mismatch between cells are hard to model and therefore ignored. The increase in mismatch for the sense amplifiers is included, however, and assumed to increase the offset mismatch standard deviation from 16.5 mV to 17.5 mV. This corresponds with the expected increase in threshold mismatch (A_{VT}) for an NMOS or PMOS input pair with a length between 40 nm to 120 nm in [33]. This requires a slight decrease of the refresh period to obtain the same cell yield.

3.4.2. Cryogenic results

Using the modelled effects mentioned previously, we can calculate new expected capacitance values and memory metrics at 4.2 K. Table 3.14 shows the memory metrics at 233 K and 4.2 K. Obviously, the area will remain the same. The latency is assumed to be the same as the threshold increase and mobility increase are assumed to cancel each other during a read operation, leading to roughly the same optimal values. Additionally, the same BER target of 10^{-6} is still used to determine the refresh period for the dynamic cells. Due to the increase in sense amplifier mismatch, the yield of the static cell design gets slightly worse. Furthermore, all write and read energies decrease slightly, but this does not change the order of the memories when sorted by those energy. Finally, the refresh rate of the dynamic cells decreases significantly, leading to a reduction in retention power. Still, the 2T NW-PR cell design has the lowest retention energy of the dynamic cells, but the 3T NW-PR design now has the highest instead of the 3T PW-PR design due to a smaller decrease in refresh rate.

Table 3.14: Comparison between memory metrics at 233 K and 4.2 K.

	2T NW-PR		3T NW-PR		3T PW-PR		6T	
Metric	233 K	4.2 K	233 K	4.2 K	233 K	4.2 K	233 K	4.2 K
Area [μm^2]	466		584		608		822	
Latency [ns]	1		0.25		1		0.15	
Cell yield	10^{-6}		10^{-6}		10^{-6}		4.75×10^{-18}	7.36×10^{-18}
Write energy [fJ]	122.08	111.37	153.24	142.52	196.98	189.32	303.88	278.24
Read energy [fJ]	183.03	160.29	150.98	142.57	157.97	147.96	191.70	177.01
Retention power [nW]	457.05	0.008 42	541.99	0.012 26	605.45	0.011 74	23.06	0
Refresh rate [Hz]	51.01 k	1.026	58.42 k	1.399	55.23 k	1.126		

Changes to the memory metrics at these temperatures lead to changes in the memory landscape over the application space. The memory landscape at 4.2 K is shown in fig. 3.22. We will list some changes with respect to the memory landscape shown in fig. 3.21.

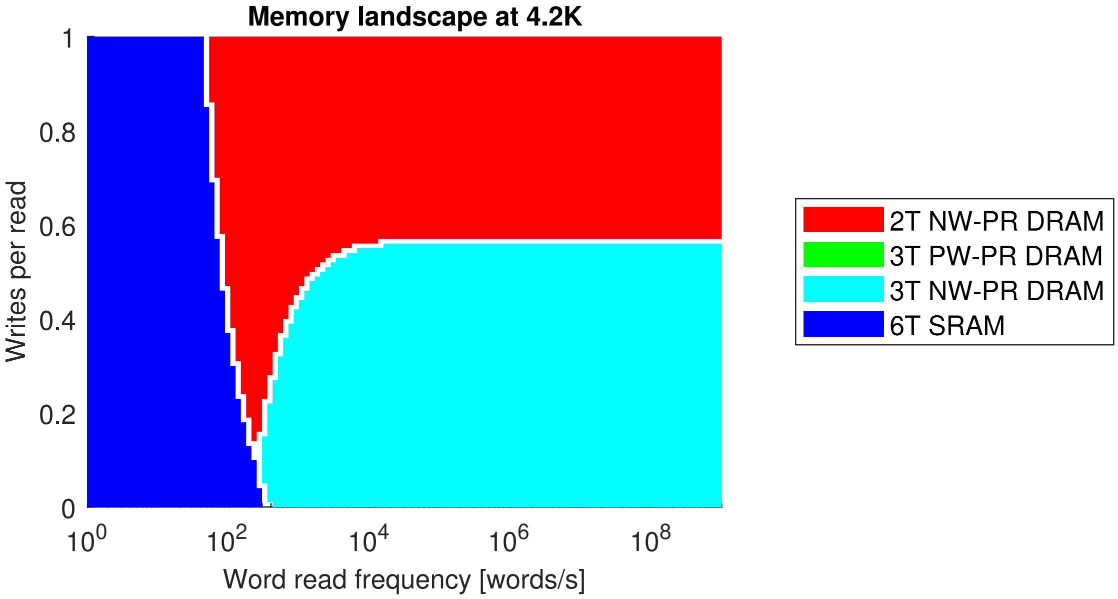


Figure 3.22: $4.2\,\mathrm{K}$ memory landscape showing the memory with the lowest power consumption over the application space with read frequencies from $1\,\mathrm{Hz}$ to $1\,\mathrm{GHz}$ and average number of writes per read between 0 and 1.

The first obvious difference is that the boundary between static and dynamic cell designs has moved to much lower frequencies. The retention power has reduced by several orders of magnitude while the read and write energies are still comparable. This means that the frequency at which the retention power starts to dominate is also several orders of magnitude lower and moves from over 10^6 to roughly 100 words read per second.

Additionally, the 2T NW-PR cell design becomes interesting for some applications with a lot of writes. Previously, the 2T NW-PR was only interesting for a specific frequency range due to its lower retention power than the 3T NW-PR cell. At $4.2\,\mathrm{K}$, the 2T NW-PR cell design becomes the best choice for applications with on average 1.75 reads per write or less, such as queues or high write activity working memories. This is due to the reduction in readout leakage that is specific to the 2T NW-PR cell, lowering the read energy such that the added read and write energy of the 2T NW-PR cell are lower than that of the 3T NW-PR cell. Again, the 3T PW-PR cell design never gives the lowest power solution, and the 3T NW-PR cell design stays the best solution for read dominated applications such as LUTs or low write activity working memories.

3.5. Model limitations and possible improvements

Although this model gives us some insight into the expected memory metrics at cryogenic temperatures, there are several shortcomings upon which to improve. We will list some limitations and improvements which could make the model more complete and allow for better visualisation of the trade-offs.

- The model does not include the latency into the memory landscape. Memories with a read and write latency of $1\,\mathrm{ns}$ can never obtain a $1\,\mathrm{GHz}$ write and read frequency. This happens in the top-right corner of the memory landscapes where the 2T NW-PR cell memory is shown to be best while it can not perform there.

  To better visualise this trade-off between the fast and slow cell designs, the latency must be included. Near the right side of the memory landscape, the read and write duration should be shortened for the slower memory cell designs. This will automatically lead to worse BER and therefore higher refresh rates and higher power consumption. As a result, the faster cell may become the lower power option again.

- The latency of the instructions is fixed based on achieving the maximum margin and reasonable bitline swings to maximise the retention time. Alternatively, a high frequency memory is not limited by its retention power. As a result, the operation durations can be reduced at the cost of an increased refresh rate, but also result in smaller bitline swings and therefore lower operational power consumption. This method may result in a completely different trade-off.

- The array size in the model is fixed. By varying the size of the memories, they may fit better in a given application and the trade-off could be slightly different. Including the size is not that difficult, since the capacitances will simply scale. However, the read operation durations will have to increase to create the same bitline swing on the increased capacitance. The bitline curves can not simply be scaled to model this and require simulation again. Additionally, effects such as the 2T NW-PR readout leakage change when changing the number of cells.
- The mismatch increase in the model only affects the sense amplifiers. In reality, the mismatch between memory cells will also increase, resulting in a larger bitline spread, worse yield and therefore more frequent refreshes. The cell mismatch is hard to model as it not only affects the readout currents generated by the readout transistor, but also the weak state voltage drop across the write pass transistor and the range of leakage currents through both transistors.
- The $4.2\,\mathrm{K}$ behaviour is based on assumptions. We can qualitatively predict the effects of cryo-CMOS characteristics on the designs, but converting quantitive cryo-CMOS characteristic changes to quantative changes in the cell characteristics is hard.
- The calculations are based purely on simulations of preliminary schematics and layouts. This means that there may be a mismatch between the model and an actual implementation if the schematics or layouts have to change.
- The models are not complete. The area calculation does not include the timing-and-coordination circuitry, the latency does not include the peripheral overhead, the yield and BER calculations only assume normal and lognormal distributions, and the power calculations only include the energy required to toggle the array lines and ignores the energy required to toggle the internal nodes of the peripherals.

3.6. Conclusion

In this chapter, we have seen the development of a memory model which can be used to investigate the trade-offs between the different memory cell designs based on area, latency, yield/BER, and power. By using data from simulations performed at $233\,\mathrm{K}$, the different designs are compared and depending on the operation frequency and read/write operation ratio the memory design with the lowest power consumption can be selected. The model is expanded by including $4.2\,\mathrm{K}$ cryo-CMOS characteristics to predict performance at $4.2\,\mathrm{K}$ which shows that the 2T NW-PR cell design becomes a better choice for a larger range of applications.

At $233\,\mathrm{K}$, the static cell design requires the lowest amount of power for almost all applications, until 10^7 operations per second. Dynamic cells only start to become interesting for higher activity. The 2T NW-PR and (mainly) the 3T NW-PR cell designs result in lower power consumption for high activity applications.

At $4.2\,\mathrm{K}$, the boundary between static and dynamic cells moves down to only 300 operations per second. As a result, dynamic cell designs become feasible for almost all applications. For write-heavy applications, the 2T NW-PR cell design results in the lowest power consumption, while the 3T NW-PR cell design gives the lowest power consumption for all other applications.

The main issue with the model is that the $4.2\,\mathrm{K}$ behaviour is based on assumptions and can not be modelled reliably. This limits the accuracy of the model and therefore the reliability of its results. To address this issue, we will design memories in chapter 4, with the cells used in the model, to be taped-out and measured. Measurement and characterisation results can then be used to calibrate and improve the model, such that it can be used to guide decisions on the best memory cell design for a given application in a cryogenic environment.

4

Memory design

In the previous chapter, we have seen how a memory model can be developed that can help us to select the best memory cell design for an application. In this chapter, we will design a number of memories that can be used to measure certain memory metrics at $4.2\,\mathrm{K}$ to validate the output of the model. Additionally, we can use the memories to obtain data that is used by the model, such as information about mismatch, noise, and bitline voltages.

This chapter is structured as follows. First, we will cover the design process for the dynamic memories by designing the cell arrays, followed by the peripherals, in section 4.1. This is followed by the design of the static memory in a similar fashion in section 4.2 which includes the design of a structure to measure the SNM of the static cell design. This chapter is concluded in section 4.3.

4.1. Dynamic memories

In the following section, we will look at the design process for the dynamic cell memories. Although the cell arrays are different, the schematics and layouts of the peripherals are similar or even reused. This ensures that differences in performance are due to the cell designs instead of optimised peripherals. Additionally, it significantly reduces design effort.

The structure of this section is as follows. First, we will look at the design of the cell arrays for each of the dynamic cell designs. This is followed by the design of the peripherals: the row decoder, write bitline drivers, sense amplifiers, and read data latches. Next, we will show the design of timing circuitry used for generating the internal pulses that control the memory operations. Finally, all components are combined to create the full memories.

4.1.1. Cell arrays

We will first look at the design of the cell arrays, as their design puts constraints on the peripherals. The cell array loads the outputs of peripherals, such as the row decoder and write bitline driver, so the array line capacitances needs to be approximately known before sizing the peripherals. Additionally, matched layout pitch is required for efficient layout and matched rows and columns.

For each cell design, two arrays with different threshold flavour devices are designed. The arrays are initially designed using mostly standard threshold devices and simulated at $233\,\mathrm{K}$ simulator temperature. Since the threshold will increase at cryogenic temperatures with approximately $100\,\mathrm{mV}$, we will copy the same design but replacing the standard-threshold devices with low-threshold devices. It is expected that the standard-threshold arrays will work better at room temperature, while the low-threshold arrays will work better at cryogenic temperatures.

2T NW-PR

We will first look at the 2T NW-PR cell, which is the smallest of the dynamic cells. It consists of only two transistors, an NMOS pass transistor for write operations (NW) and a PMOS transistor for readout (PR). The schematic and further operational details of this cell type can be found in section 2.1.2.

The write pass NMOS transistor is sized to be a minimum size transistor with a W/L of $120\,\mathrm{nm}/40\,\mathrm{nm}$. Due to short channel effects, increasing the length will increase the threshold voltage and therefore re-

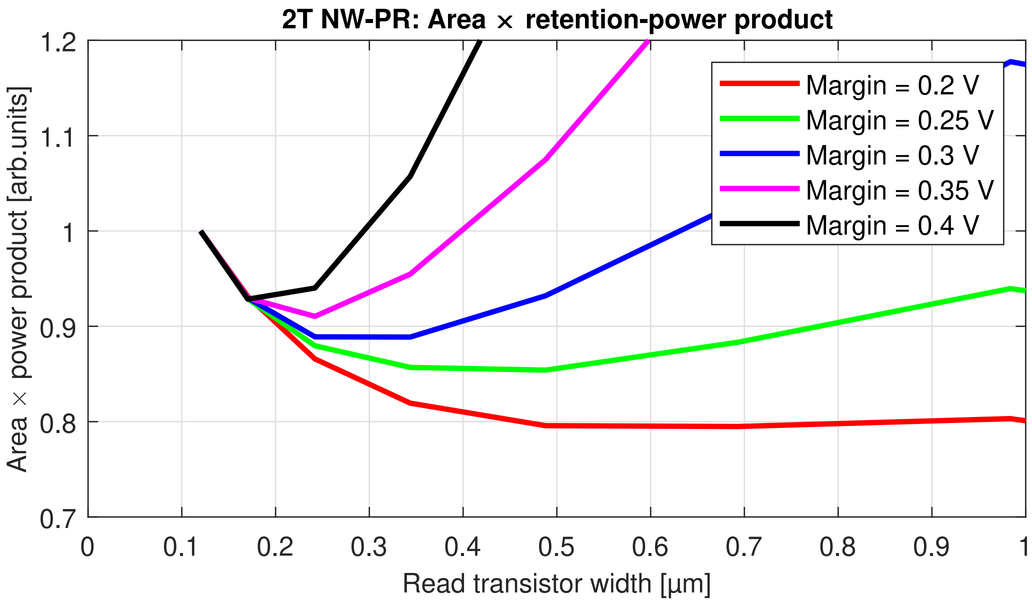


Figure 4.1: 2T NW-PR cell area×retention-power product as a function of the read transistor width.

duce the highest voltage that can be written to the storage node. Increasing the width will allow for a faster write, but also increase the leakage current, which will limit the retention time, and increase the write wordline capacitive load and therefore the write operation energy. As shown in the simulation results in fig. 3.4, a minimum width already results in a $10\,\mathrm{ps}$ strong write duration and weak write exponential settling within $100\,\mathrm{ps}$. Finally, a minimum size write transistor will also help in achieving the smallest cell area possible.

The readout PMOS transistor is sized with a W/L of $300\,\mathrm{nm}/40\,\mathrm{nm}$. The transistor is sized by comparing the bitline margins of read operations for various widths and lengths after a certain hold time. A minimum length always results in the largest margins, so a minimum length transistor is used. This results in an area-retention trade-off where a larger width increases the total area, but also increases the storage capacitance and therefore increases the retention time.

The width of the read transistor is determined by selecting the size that maximises the area×retention-power product. This is determined by simulating the bitline margin of a read operation at various hold times and read transistor widths. The retention power is inversely proportional to the retention time which is defined as the hold time at which the nominal bitline margin becomes less than a certain threshold. The area is proportional to the width of the read transistor added to a constant width which is needed for the write transistor and required spacing overhead. Figure 4.1 shows these normalised values for a range of widths for various minimal nominal bitline margins. For nominal bitline margins around $300\,\mathrm{mV}$, which is required to obtain a high enough yield according to fig. 3.9, we find an optimal read transistor width of around $300\,\mathrm{nm}$.

The layout of a tileable double-cell is shown on the left side of fig. 4.2 (distorted to protect exact dimensions). The write pass transistors are at the top and bottom, and the readout transistors are in the middle with a shared read wordline in the shared active region. In metal layer 2, there are two horizontal read bitlines connected to the two readout transistors, and two horizontal write bitlines connected to the write pass transistors. The wordlines run vertically in metal layer 3, and are shared between the two cells. This layout therefore corresponds to a single row with two columns. Note that the cell layout is rotated $90°$ with respect to the cell arrays shown previously (figs. 2.1 and 2.2).

The bitlines run on the second metal layer to improve their line capacitance. First of all, the second metal layer runs horizontally, in the smaller cell dimension. This leads to a lower per cell capacitance than vertical lines, which is important for bitlines since on average sixteen of them are toggled during an operation, compared to only a single wordline (shown in eqs. (3.4) and (3.5), and eqs. (3.7) and (3.8)). The bitlines are also covered by the wordlines and therefore their total capacitance is not influenced by the metal patterns on layer 4 and above. This also shields the bitlines from lines running over the memory which could affect the bitline voltages during readout. This makes the readout more reliable.

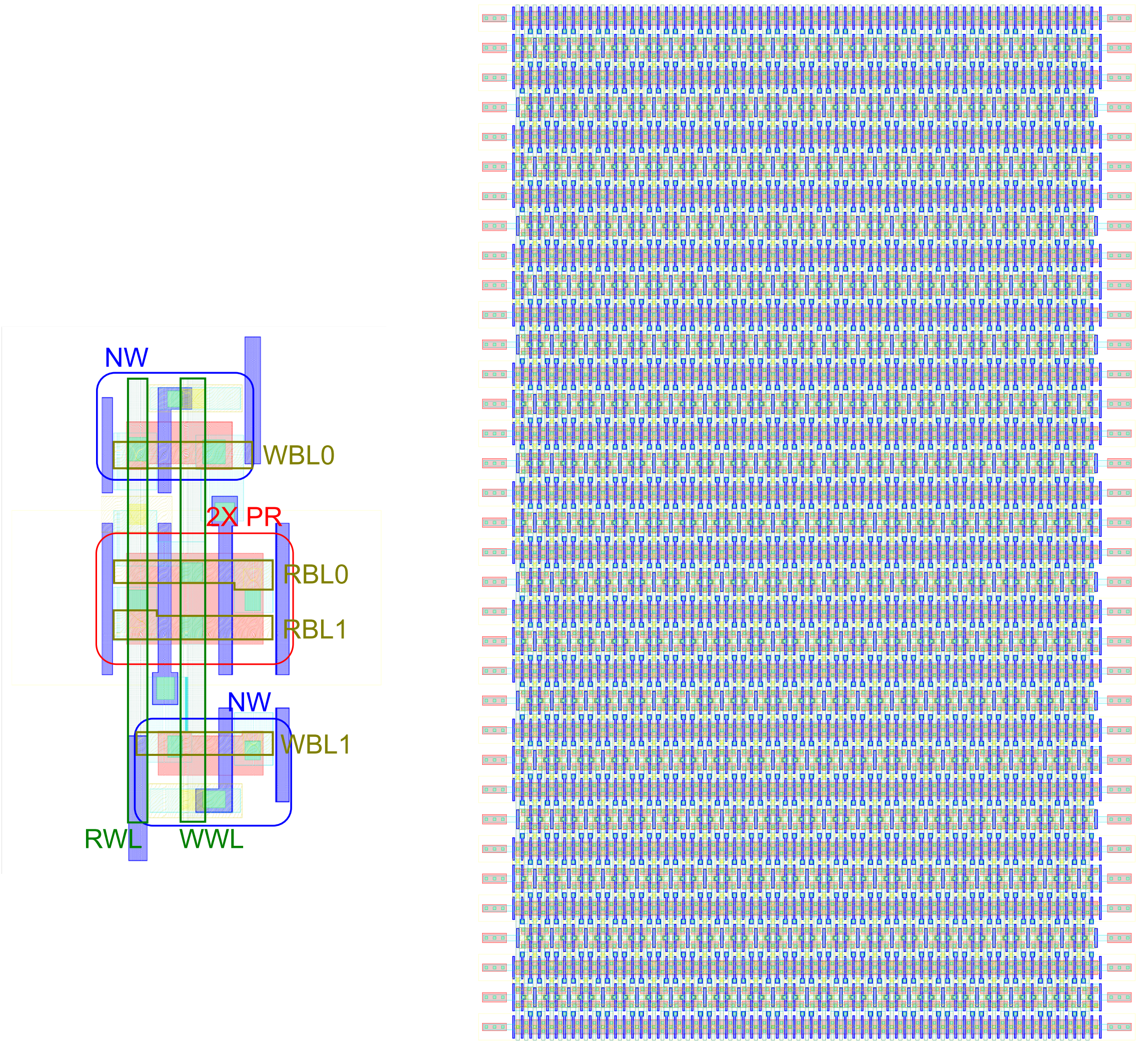


Figure 4.2: Left: (distorted) layout of a pair of 2T NW-PR memory cells with highlighted NMOS write transistors (NW), PMOS read transistors (PR), and wordlines (WL) and bitlines (BL) for read and write operations on both cells. Right: layout of 2T NW-PR memory cell array with 32 rows, 32 columns, dummies, and well taps. See appendix A for layer legend.

The shown layout is tileable when combined with a layout that is mirrored across the horizontal axis. These two layouts can be alternated vertically and share the write wordline contact vias. Additionally, these layouts can be alternated horizontally and share read bitline contact vias. This results in an effective layout size of $1.26\,\mu\mathrm{m}$ by $0.36\,\mu\mathrm{m}$ for two cells, and a cell area of $0.2268\,\mu\mathrm{m}^2$.

The cell design is tiled into an array, which is surrounded by a ring of dummy cells and two columns of well taps. The ring of dummy cells ensures that cells near the edge of the array and in the middle of the array match, by replicating the layout environment. The columns of well taps ensure that the substrate potentials are set. This becomes harder at cryogenic temperatures due to freeze-out which increases the substrate resistance significantly. This is not expected to cause issues for the array, since the largest distance between a transistor and the corresponding well taps is $6.36\,\mu\mathrm{m}$. The layout of the full array is shown on the right side of fig. 4.2 and consumes an area of $306\,\mu\mathrm{m}^2$.

3T NW-PR

The 3T NW-PR cell design is very similar to the 2T NW-PR cell design, but includes the additional transistor in the readout stack. Effectively, it is a 2T cell which is always in readout mode, but the readout current is switched using an additional read pass transistor. The schematic and further operational details of this cell type can be found in section 2.1.2.

The sizing of the 3T NW-PR cell is similar to that of the 2T NW-PR cell. The sizes for the write pass and readout transistors are equal to those for the 2T NW-PR cell: $120\,\mathrm{nm}/40\,\mathrm{nm}$ and $300\,\mathrm{nm}/40\,\mathrm{nm}$, respectively. The read pass transistor is made as wide as allowed by the layout rules to minimise its on-resistance without increasing the total cell area. This results in a W/L of $190\,\mathrm{nm}/40\,\mathrm{nm}$.

Since there is no coupling between the read wordline and the storage node, the written storage-node voltage is used directly for readout, contrary to the 2T NW-PR design where the storage-node voltage is raised during readout. Due to the threshold voltage drop across the N-type write pass transistor during a write-1 operation, the P-type readout transistor can not be fully put into cutoff and a read operation will always cause an increase in the bitline voltage, resulting in a higher read operation energy consumption. To combat this, the threshold of the readout stack transistors is increased, by using high threshold voltage devices. This will slightly slow down the read after a write-0 operation, but prevent charging of the bitlines after a write-1 operation. For the low-threshold variant of the cell array, the readout stack is implemented using standard threshold voltage devices instead and the write transistor is implemented using a low threshold voltage device.

The layout of a tileable cell is shown on the left side of fig. 4.3 (distorted to protect exact dimensions). The N-type write pass transistor is at the top, and the P-type readout stack is at the bottom. Again, the wordlines run vertically along the larger cell dimension on the third metal layer, and the bitlines run horizontally along the smaller cell dimension on the second metal layer.

The cell design can be tiled into an array by mirroring across the cell boundary. The contact vias are shared between neighbouring cells for reduced layout size. The resulting effective cell size is $0.83\,\mu\mathrm{m}$ by $0.36\,\mu\mathrm{m}$, or a cell area of $0.2988\,\mu\mathrm{m}^2$.

Similar to the 2T NW-PR cell array, the single cell layout is tiled into an array and surrounded by dummy cells and well taps. The resulting array layout is shown on the right side of fig. 4.3 and consumes a total area of $369\,\mu\mathrm{m}^2$.

3T PW-PR

Finally, the 3T PW-PR cell design is similar to the 3T NW-PR cell design, but only features P-type devices and has an additional wordline connection. Due to the data-dependent coupling capacitance, the storage node voltage difference is amplified during readout. The schematic and further operational details of this cell type can be found in section 2.1.2.

The sizing of this cell is identical to the sizing of the 3T NW-PR cell design, and the layout is similar, as shown on the left side of fig. 4.4 (distorted to protect exact dimensions). However, in the 3T NW-PR cell design, the source of the readout transistor is connected to the supply voltage. This means that this node can be shared with any neighbouring cell, regardless of its position in the memory array. In the case of the 3T NW-PR cell design, the node is shared horizontally with cells in the same column. Since the source of the readout transistor is now connected to the read wordline, the neighbouring cell at the readout transistor source side must be in the same row, as it uses the same wordline. This means that a group of four cells physically arranged as a 2×2 cell group functionally behaves as a single row and four columns.

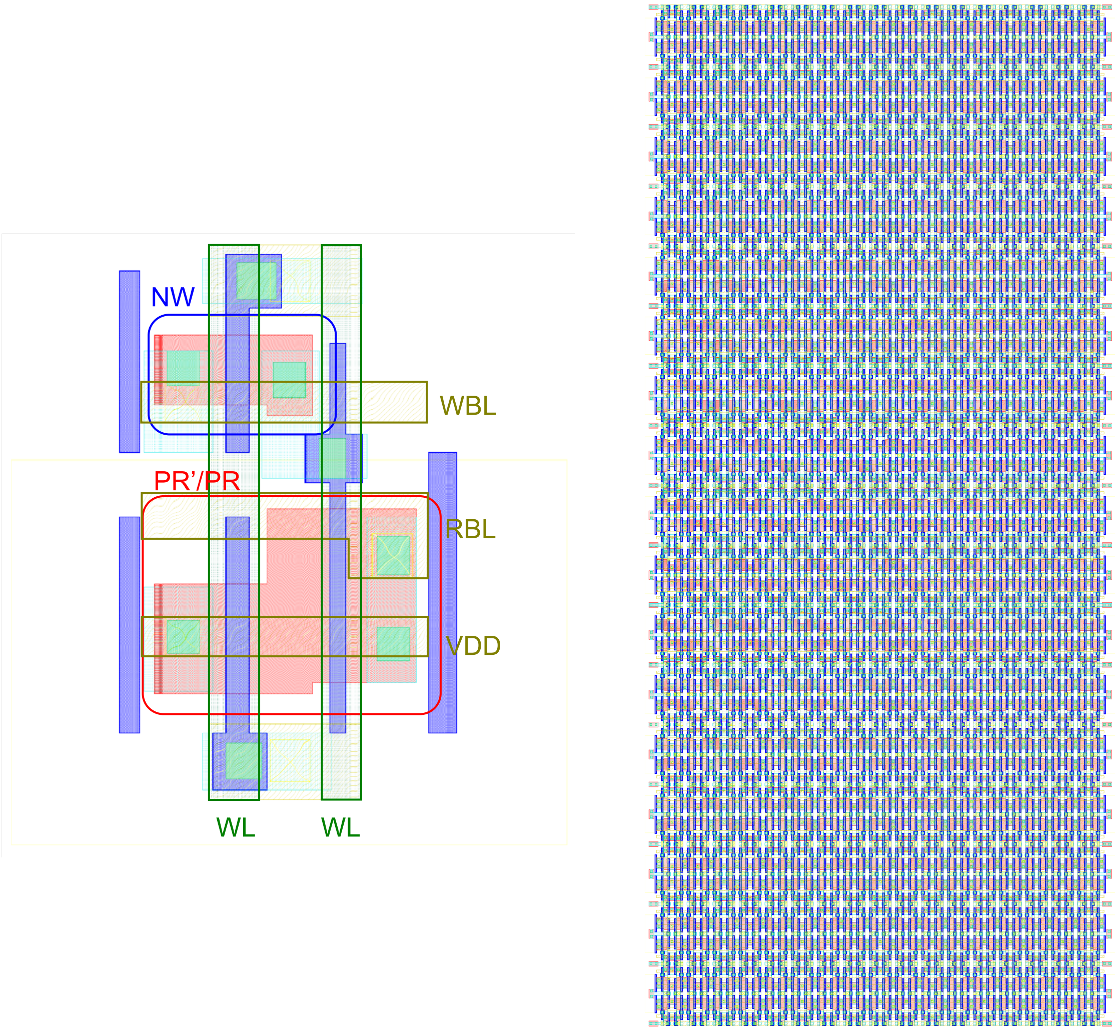


Figure 4.3: Left: (distorted) layout of a 3T NW-PR memory cell with highlighted NMOS write transistor (NW), PMOS read transistors (PR/PR'), and wordlines (WL) and bitlines (BL) for read and write operations. Read/write assignment for WLs depends on the cell orientation in the array. Right: layout of 3T NW-PR memory cell array with 32 rows, 32 columns, dummies, and well taps. See appendix A for layer legend.

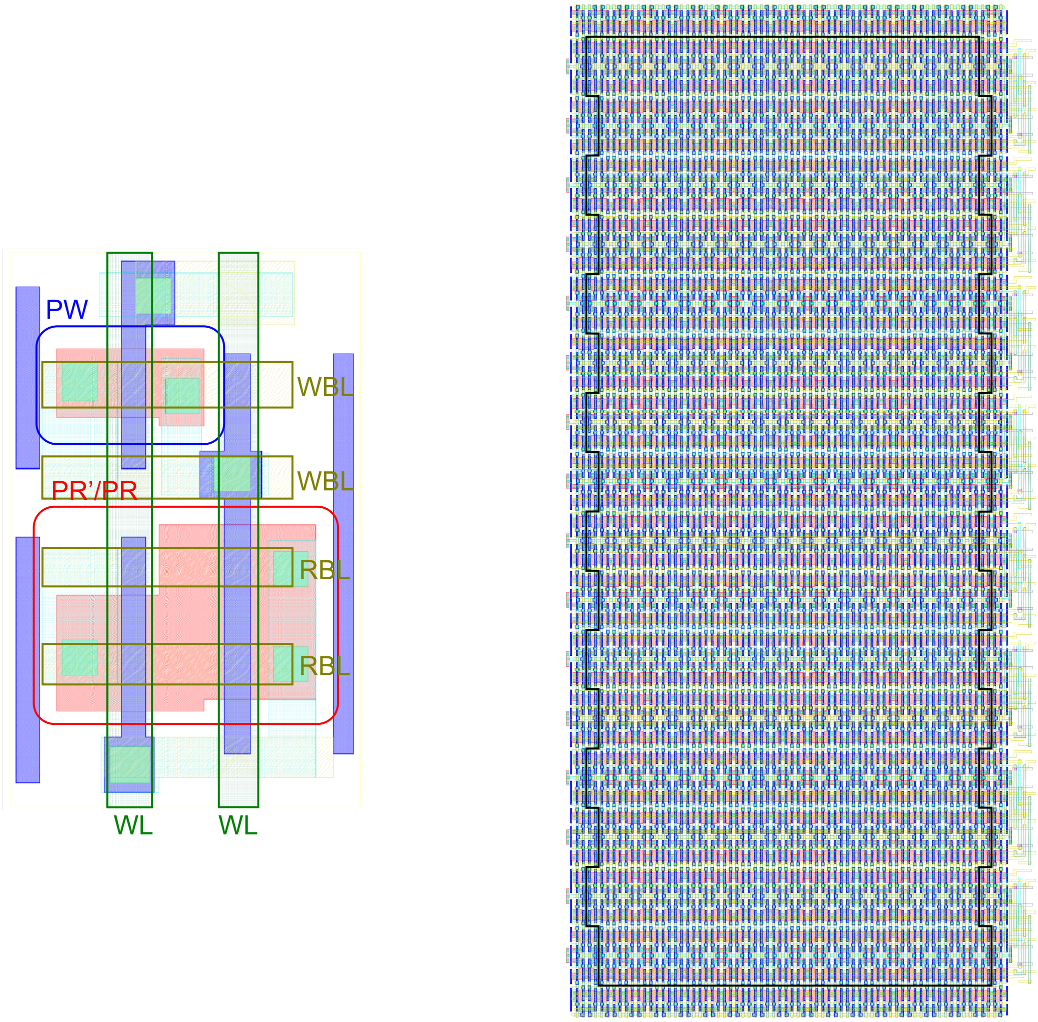


Figure 4.4: Left: (distorted) layout of a 3T PW-PR memory cell with highlighted PMOS write transistor (PW), PMOS read transistors (PR/PR'), and wordlines (WL) and bitlines (BL) for read and write operations. Read/write assignment for WLs and the selection of the BLs from the BL pairs depends on the cell orientation in the array. Right: layout of 3T PW-PR memory cell array with 32 rows, 32 columns, dummies, and well taps. The black line outlines the functional memory cells and shows the staggered column pattern. See appendix A for layer legend.

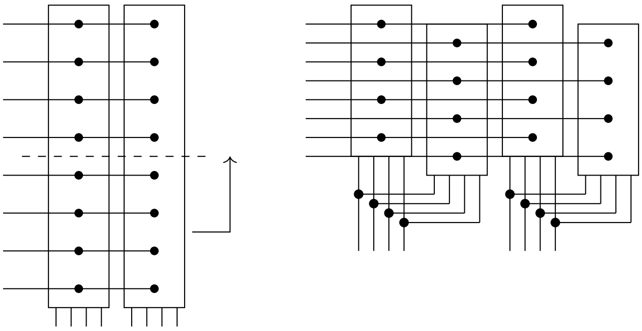


Figure 4.5: Interleaving of 3T PW-PR columns to shorten bitlines and maintain the desired cell array aspect ratio. Example shows eight horizontal wordlines connected to two columns with four vertical bitlines each. The columns are cut and interleaved to double the wordline length and halve the bitline length.

The horizontal wordline sharing issue leads to a high power consumption and layout issues if not addressed. The height of a row will double, while the width of a column halves. This results in a cell array which is long in the bitline direction and short in the wordline direction. This will halve the wordline capacitance, but double the bitline capacitance. Since bitline capacitance contributes a large part of the power consumption, this will increase the power consumption significantly. Additionally, the effective bitline pitch is halved which results in impossible bitline peripheral layout.

The issue is solved by interleaving columns that do not share wordlines. For each column, a second column is added with a vertical offset equal to a single cell height. This creates a staggered column layout where the wordlines are not shared between the two neighbouring columns as shown in fig. 4.5. However, bitlines are now shared between two columns and the total capacitance per bitline has not halved. This is accepted for the write bitlines since there is no space for double the amount of write bitline drivers. However, it is not acceptable for the read bitlines as they will significantly slow down the speed of readout. Hence, the read bitlines are not connected together directly and instead the desired bitline from each pair is selected for a read operation by a single sense amplifier.

The cell layout is again tiled into an array and surrounded by dummy cells and well taps. The right side of fig. 4.4 shows the layout of the resulting array, where the staggered column layout is clearly visible in the black outline which surrounds the functional memory cells, not including dummy cells and well taps. On the right side of the array, the bitlines from the different columns are combined. The write bitlines are connected together, and the read bitlines are grouped such that corresponding bitlines lie close to each other.

4.1.2. Row decoder

To select a single row based on an address, we need to have a row decoder. It has to decode a $5\,\mathrm{bit}$ binary address into a one-hot $32\,\mathrm{bit}$ signal. For the dynamic cell arrays, we need two row decoders: one for the write wordlines and one for the read wordlines. All dynamic cell arrays have wordline pitch of $360\,\mathrm{nm}$, which means that the pitch of the layout must also be $360\,\mathrm{nm}$. Additionally, some of the memories require the wordlines to be all high, except the selected one. This can be achieved by adding a single layer of inverters to the outputs of a decoder.

The row decoders uses a dynamic decoder design with predecoding. A standard dynamic decoder design is shown in fig. 4.6. When the clock ϕ is low, the dynamic node is precharged to the supply. When the clock ϕ is high, the dynamic node either stays high or is discharged through the pull-down stack. By connecting the pull-down transistors to each address bit or its inverse, the dynamic node is discharged for a single address only. This type of design requires less transistors than a static design, uses less switching nodes, and is inherently glitchless.

A predecoder is used to further reduce the number of transistors and power consumption, as shown in fig. 4.7. Instead of using four signal lines for two address bits and their inverse, the four signal lines are used for a one-hot decoded version of those two bits. This allows us to use only a single transistor in the pull-down stack for two address bits. This method is used for the four most significant bits, leading to only two transistors in the pull-down stack. Additionally, the footer transistor is removed and instead the least significant bit (LSB) address line and its inverse are gated by the clock. This also allows for sharing of the bottom two transistors of the pull-down stack between neighbouring decoder sections.

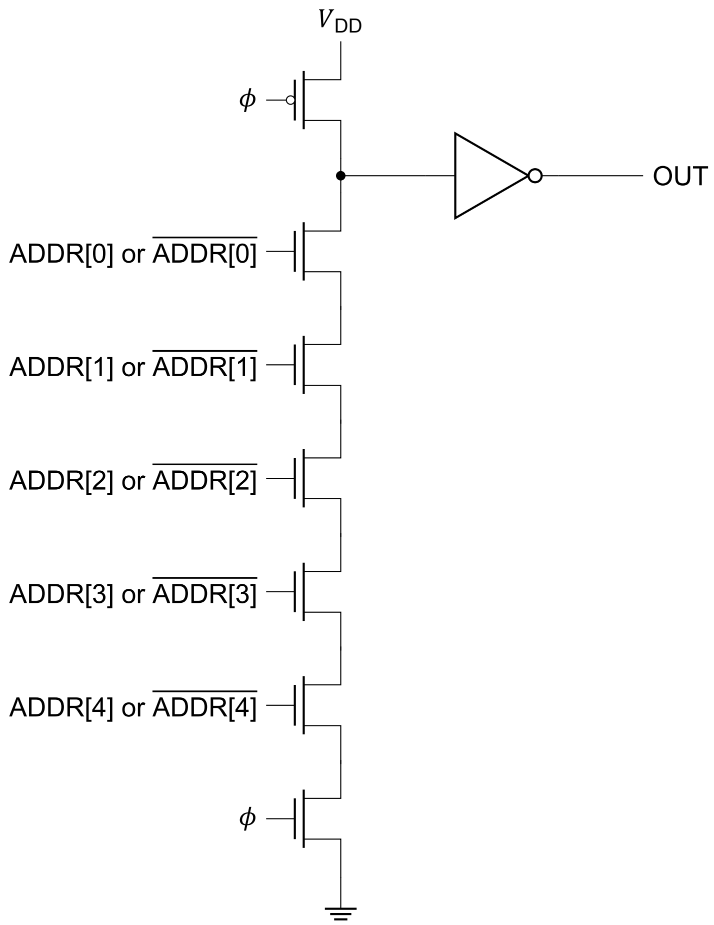
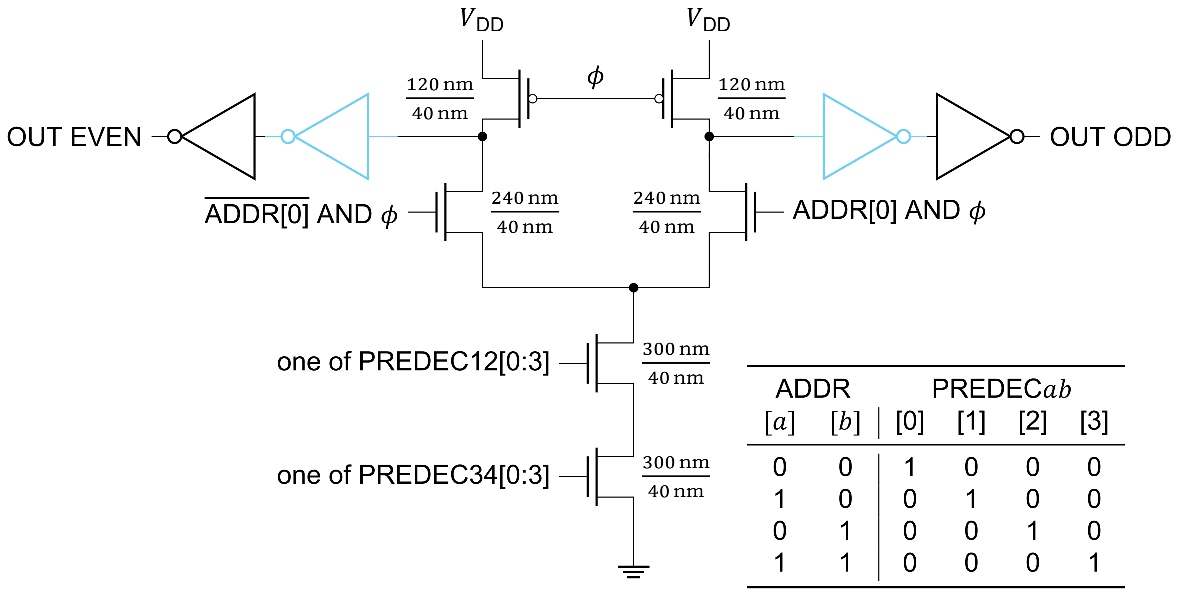

V_{DD}
ϕ
OUT
ADDR[0] or $\overline{\text{ADDR[0]}}$
ADDR[1] or $\overline{\text{ADDR[1]}}$
ADDR[2] or $\overline{\text{ADDR[2]}}$
ADDR[3] or $\overline{\text{ADDR[3]}}$
ADDR[4] or $\overline{\text{ADDR[4]}}$
ϕ

Figure 4.6: Schematic for a single output of a 5 bit dynamic decoder. The address of the output is encoded in selection of the address bits or their complements in the pull-down stack.


V_{DD}
V_{DD}
ϕ
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
OUT EVEN
OUT ODD
$\overline{\text{ADDR[0]}}$ AND ϕ
ADDR[0] AND ϕ
$\frac{240\,\mathrm{nm}}{40\,\mathrm{nm}}$
$\frac{240\,\mathrm{nm}}{40\,\mathrm{nm}}$
one of PREDEC12[0:3]
$\frac{300\,\mathrm{nm}}{40\,\mathrm{nm}}$
one of PREDEC34[0:3]
$\frac{300\,\mathrm{nm}}{40\,\mathrm{nm}}$

ADDR		PREDECab			
$[a]$	$[b]$	[0]	[1]	[2]	[3]
0	0	1	0	0	0
1	0	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

Figure 4.7: Schematic of a row decoder section for a wordline pair with transistor sizing. The PMOS/NMOS sizing of the final inverters is $\frac{1.2\,\mu\mathrm{m}}{40\,\mathrm{nm}}/\frac{0.6\,\mu\mathrm{m}}{40\,\mathrm{nm}}$. The PMOS/NMOS sizing of the cyan inverters inserted for the one-low output design is $\frac{320\,\mathrm{nm}}{40\,\mathrm{nm}}/\frac{240\,\mathrm{nm}}{40\,\mathrm{nm}}$. The table shows how two address bits will get predecoded into a one-hot signal, allowing a shared NMOS stack of only two transistors.

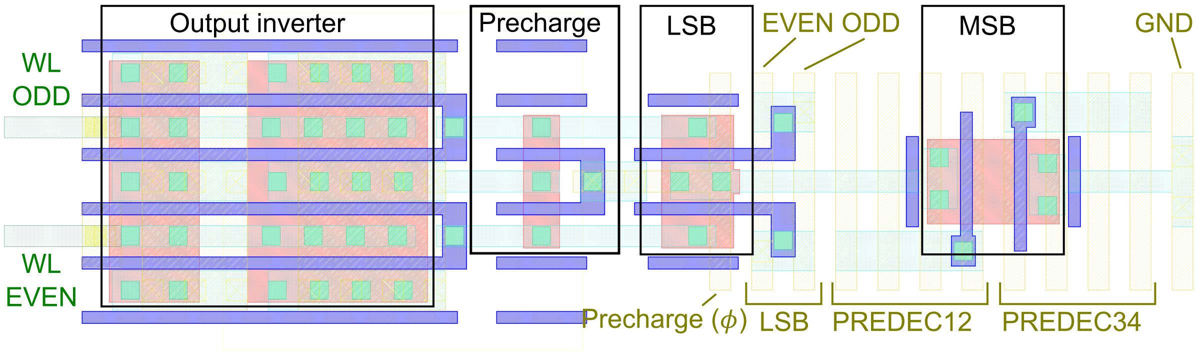


Figure 4.8: Layout of a row decoder section for a wordline pair with one-high output. The gates of the MSB transistors are connected to the PREDEC lines when all sections are combined to form the full array. See appendix A for layer legend.

We will first look at the sizing and layout of the LSB transistor, precharge transistor, and driver inverters which should be fast enough to drive the wordlines and also fit within the row pitch of the cell array. The precharge transistor is strong enough at minimum size (W/L = $120\,\mathrm{nm}/40\,\mathrm{nm}$), and the LSB transistor is sized at double minimum width and minimum length (W/L = $240\,\mathrm{nm}/40\,\mathrm{nm}$) to speed up the discharge of the dynamic node through the entire pull-down stack. The driver inverter is sized such that the wordlines are toggled quickly, much faster than the duration of a clock (ϕ) pulse. Additionally, the inverter must be able to provide readout current in case of the 2T NW-PR and 3T PW-PR cells designs. This leads to a W/L of $1.2\,\mu\mathrm{m}/40\,\mathrm{nm}$ for the PMOS transistor and $0.6\,\mu\mathrm{m}/40\,\mathrm{nm}$ for the NMOS transistor.

The inverted output decoder uses two inverters between the dynamic node and the output of the decoder. The final inverter is sized the same as the final inverter of the regular decoder. The intermediate inverter has a PMOS W/L of $320\,\mathrm{nm}/40\,\mathrm{nm}$ and an NMOS W/L of $240\,\mathrm{nm}/40\,\mathrm{nm}$.

The two NMOS transistors in the stack that are driven by predecoders are sized such that they will not significantly limit the evaluation speed and not limit the layout pitch. This leads to a W/L for both transistors of $300\,\mathrm{nm}/40\,\mathrm{nm}$. These are connected to the LSB pair and the resulting layout is shown in fig. 4.8.

Finally, the predecoder is designed using minimum sized static logic gates. The inset in fig. 4.9 shows the schematic of a $2\,\mathrm{bit}$ predecoder at the logic gate level. For the full decoder, two of these are used, together with some logic for gating the LSB with the clock as shown in the top part of fig. 4.9. Only the output inverters for the LSB are sized to $4\times$ and $2\times$ minimum width for the PMOS and NMOS transistors, respectively. In our application, the addresses will be applied well before the clock signal, so only the signal path containing the clock is timing critical.

The separate decoder sections can be stacked into an array and connected to the predecoder. Figure 4.10 shows the resulting layout, including the predecoder on the right side and added well taps. The complete decoder has a height of $3.7\,\mu\mathrm{m}$ and width of $15.1\,\mu\mathrm{m}$. All transistors are implemented using low-threshold devices to speed up the room temperature operation, and at $4.2\,\mathrm{K}$ the speed will be even higher. Note that the leakage of the low-threshold devices is higher than that of the standard-threshold devices at room temperature due to the finite subthreshold slope, which results in a trade-off between speed and leakage power. However, the leakage difference at $4.2\,\mathrm{K}$ is expected to be much smaller as it is no longer dominated by subthreshold leakage but by low activation energy leakage phenomena such as tunnelling currents. The increase in speed will reduce the peripheral latency overhead required while increasing the leakage only slightly.

The complete decoder is simulated to determine the latency between the clock and the wordline transition. Figure 4.11 shows the clock and selected wordline in a simulation including layout parasitics at $233\,\mathrm{K}$ using low-threshold devices. The resulting rising and falling edge delay are $131\,\mathrm{ps}$ and $119\,\mathrm{ps}$, respectively, in the typical process corner and should be taken into account during the design of the timing signal generation circuits. Table 4.1 shows the rising and falling edge delay for all process corners.

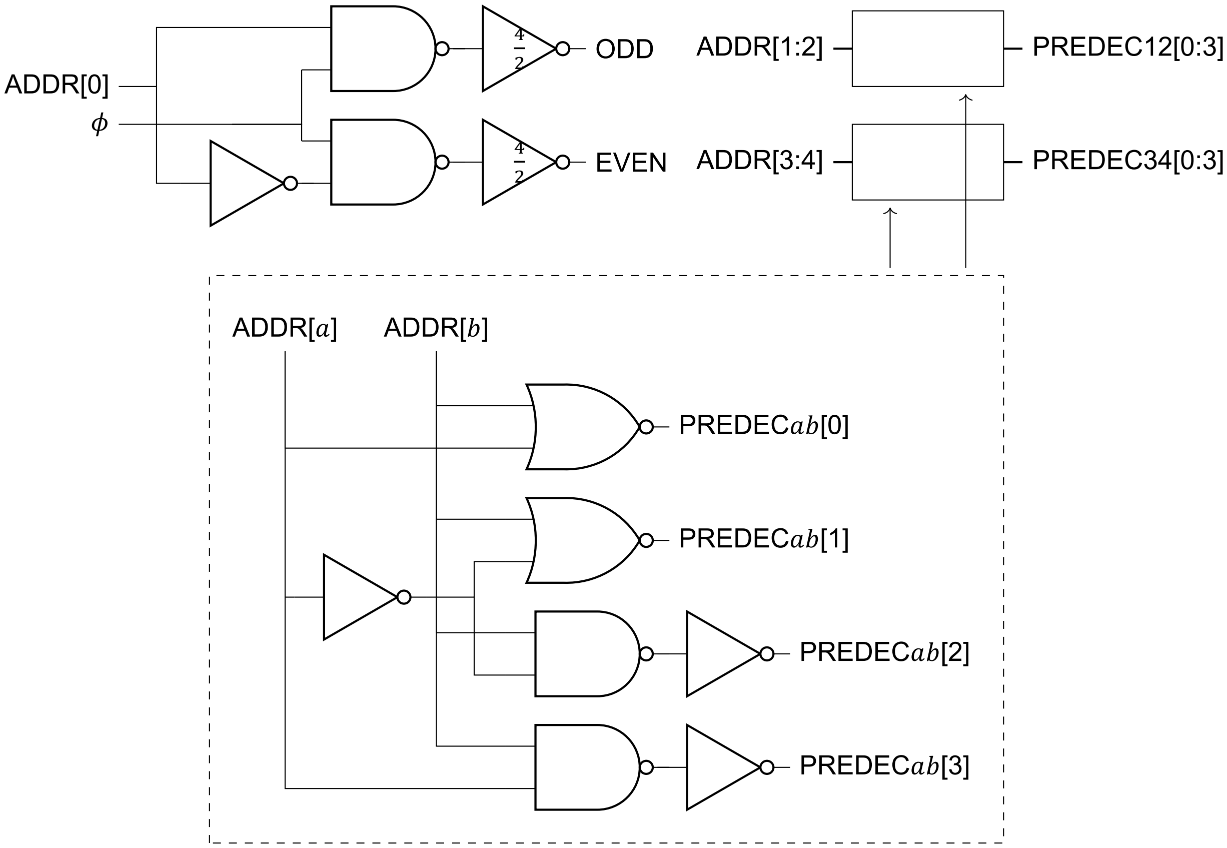


Figure 4.9: Schematic of the 5 bit predecoder. The inset shows the schematic of the 2 bit predecoder used for the four most significant bits.

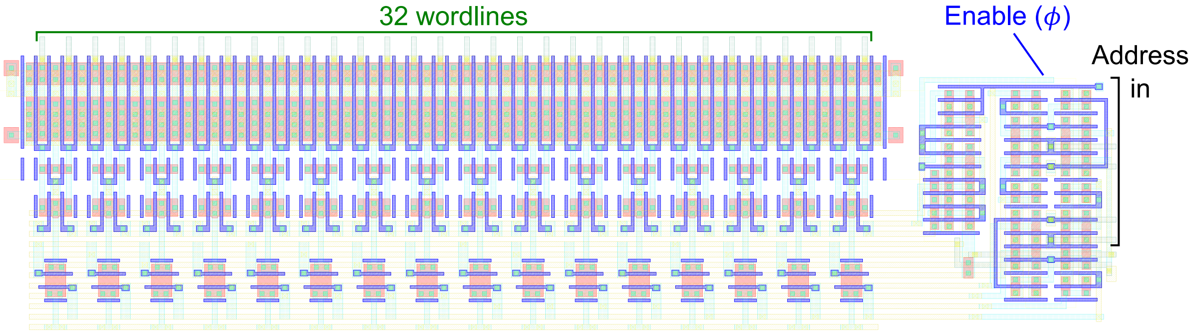


Figure 4.10: Layout of a row decoder with predecoder (right) and well taps. See appendix A for layer legend.

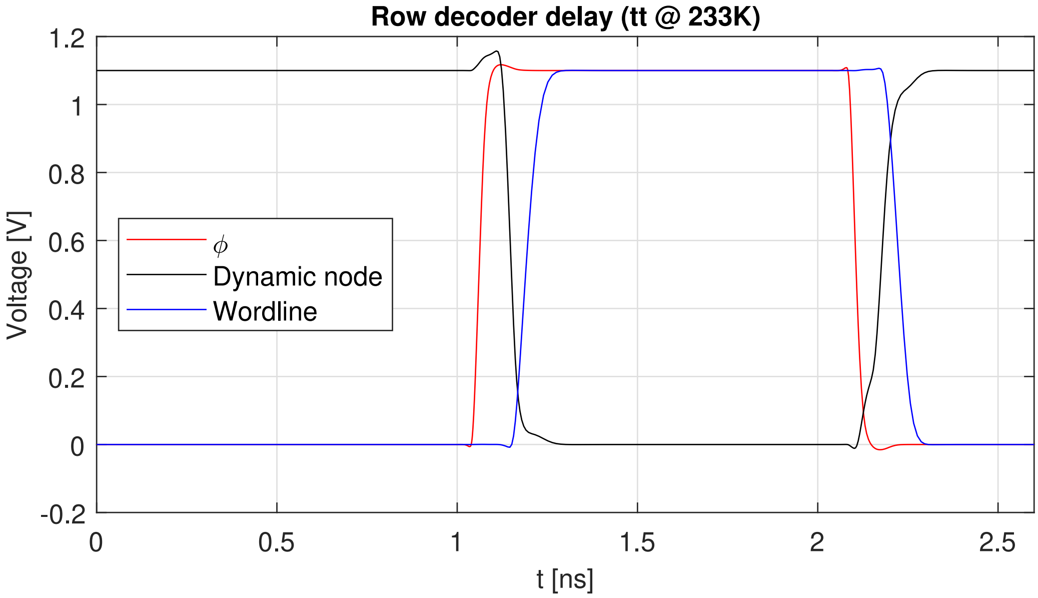


Figure 4.11: Input clock, internal dynamic node, and output wordline voltages of the complete row decoder, including layout parasitics, in the tt process corner at $233\,\mathrm{K}$.

Table 4.1: Row decoder latency across process corners.

Corner	tt	ss	ff	fs	sf
Rising edge $[\mathrm{ps}]$	131	182	96	128	135
Falling edge $[\mathrm{ps}]$	119	161	91	126	115

4.1.3. Write bitline driver

To write data into the cells, we need drivers for the write bitlines. These drivers need to either use the data applied externally for a write operation, or use the data from the sense amplifiers in case of a refresh write operation.

The write bitline driver is designed using a static logic gate made from minimum size transistors for input selection and a larger bitline driving inverter. The schematic for a single bitline is shown in fig. 4.12. The static gate on the left side implements the logic function shown in eq. (4.1). The W_{IN} and W_{REF} signals are driven using complementary signals, which means that output is either $\overline{D_{\text{IN}}}$ or $\overline{D_{\text{REF}}}$ as expected, resulting in a 2-to-1 multiplexer with inverted output. The output inverter is sized with a minimum length and $4\times$ and $2\times$ minimum width PMOS and NMOS, respectively.

$$F = \overline{W_{\text{IN}} \cdot D_{\text{IN}} + W_{\text{REF}} \cdot D_{\text{REF}}} \tag{4.1}$$

The write bitline driver for the 3T PW-PR design is slightly altered. First of all, the output inverter size is increased by a factor 2, since the write wordlines of the staggered columns are connected together, therefore roughly doubling the capacitive load. An additional inverter with double minimum width transistors is also added between the static logic gate output and the input of the driver inverter. This improves the edge steepness at the driver inverter input and inverts the data to be written, which makes the remainder of the dynamic cell memory designs more similar since the 3T PW-PR readout is non-inverting.

The width of the write bitline driver must match the bitline pitch of the cells for efficient layout. Since the pitches of the cells are all different, three different layouts are used as shown in fig. 4.13. However, large parts of the designs are reused and stretched to fit the different bitline pitches. This minimises additional design effort and keeps the peripherals as similar as possible.

Finally, the drivers can be stacked into an array with well taps at either end, similar to the stacking of sections for the row decoder. This results in three arrays with a height of $2.8\,\mu\mathrm{m}$ for the 2T and 3T NW-PR designs and $3.8\,\mu\mathrm{m}$ for the 3T PW-PR design. Due to the careful pitch matching, they fit exactly on

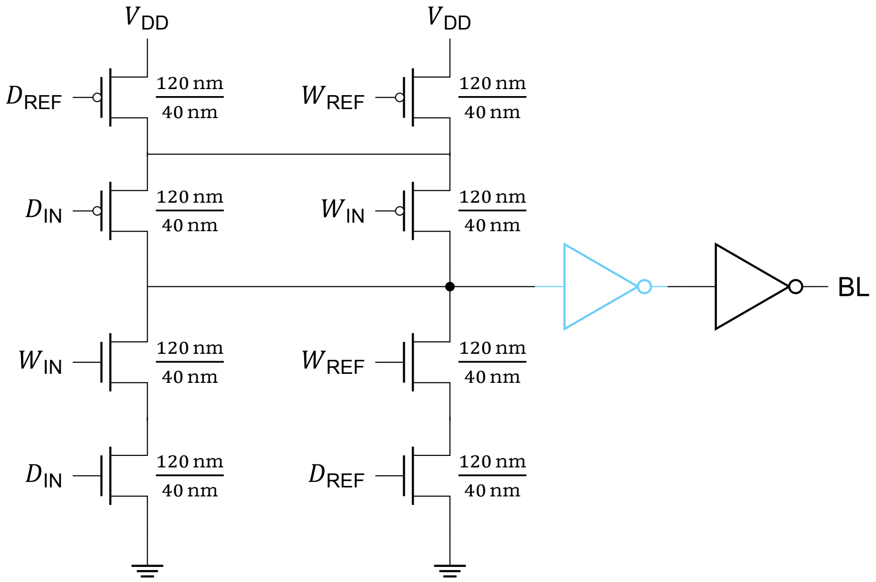

V_{DD}
V_{DD}
D_{REF}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
W_{REF}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
D_{IN}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
W_{IN}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
BL
W_{IN}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
W_{REF}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
D_{IN}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$
D_{REF}
$\frac{120\,\mathrm{nm}}{40\,\mathrm{nm}}$

Figure 4.12: Schematic of the write bitline driver with transistor sizing. For the 2T and 3T NW-PR memory designs, only the final inverter is used with a PMOS/NMOS sizing of $\frac{480\,\mathrm{nm}}{40\,\mathrm{nm}}/\frac{240\,\mathrm{nm}}{40\,\mathrm{nm}}$. For the 3T PW-PR memory design, the cyan inverter is added with a PMOS/NMOS sizing of $\frac{240\,\mathrm{nm}}{40\,\mathrm{nm}}/\frac{240\,\mathrm{nm}}{40\,\mathrm{nm}}$. The second inverter then has a PMOS/NMOS sizing of $\frac{960\,\mathrm{nm}}{40\,\mathrm{nm}}/\frac{480\,\mathrm{nm}}{40\,\mathrm{nm}}$.

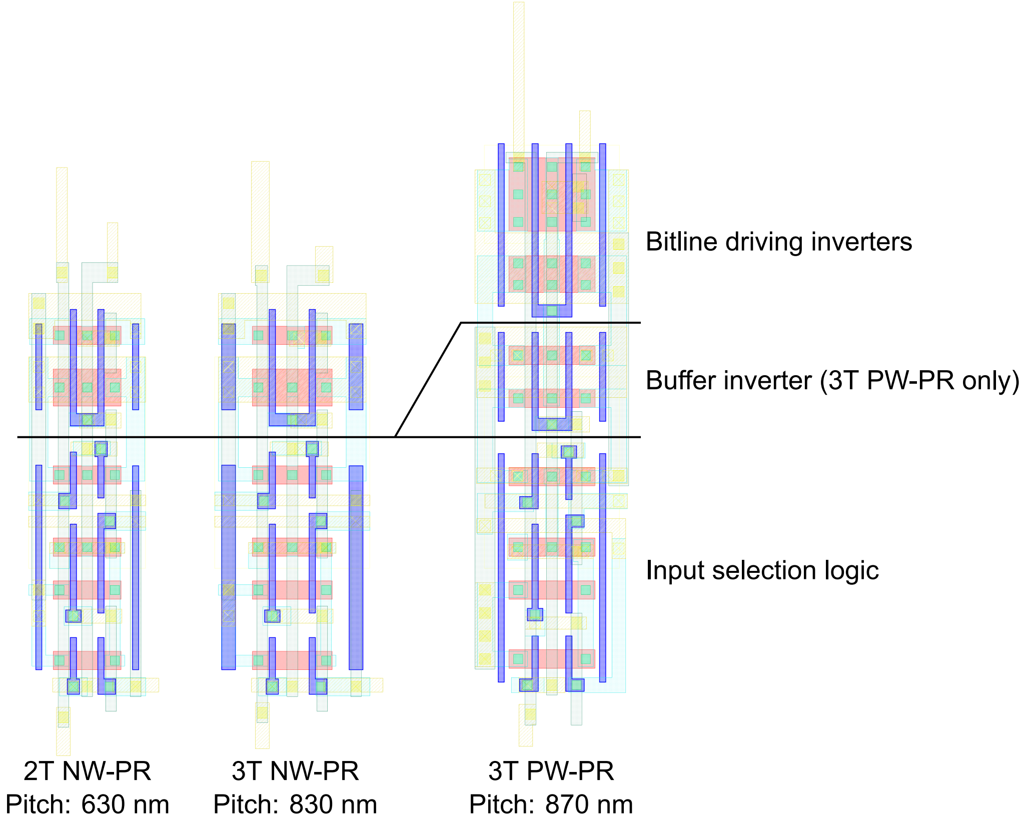


Figure 4.13: Layout of a write bitline driver section for each dynamic cell design. See appendix A for layer legend.

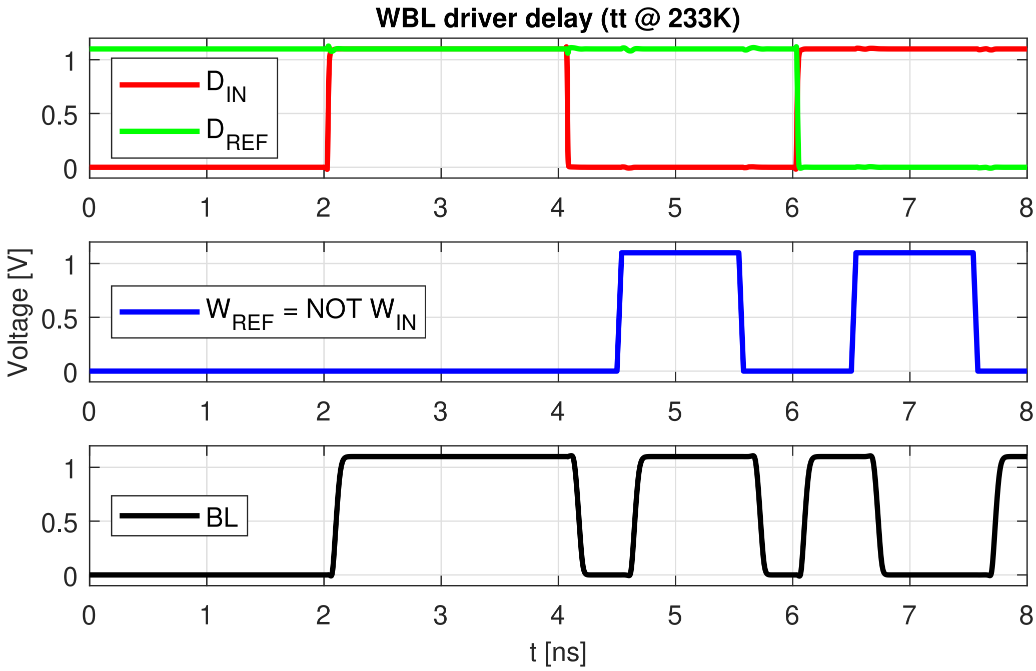


Figure 4.14: Input data, selection signal, and output bitline voltage simulation of the 2T NW-PR write bitline driver with layout parasitics in tt corner at 233 K.

the cell arrays. All transistors are implemented using low-threshold devices to improve speed at both room temperature and cryogenic temperatures.

The bitline driver is simulated to determine its latency from the inputs to the bitline voltage. It must be fast enough to switch the bitlines before the row decoder enables the write pass transistors. If it is too slow, the write duration is shortened and the data could be less reliable than expected. Figure 4.14 shows the responses of the decoder with a realistic bitline load when simulated at 233 K. The speed at 4.2 K is expected to be higher.

The input-data-to-bitline delay and refresh-selection-to-data delay over the different process corners is shown in table 4.2. During a regular write operation, the bitlines are set faster than the row decoder can decode an address, so the write operation duration will not be shortened. During a refresh write, the delay of charging the bitline is comparable to the row decoder latency in all corners. Since the latency of the write timing pulse generation circuits needs to be added to the row decoder latency, the bitlines can be charged fast enough for a refresh. Discharging is significantly slower than the row decoder, but since a write to 0 V is finished within 10 ps (see fig. 3.4), shortening of the write operation duration does not influence the storage node voltage.

Table 4.2: Write bitline driver latency across process corners.

Corner	tt	ss	ff	fs	sf
D_{IN} to BL rising [ps]	70.4	98.7	51.9	69.4	74.6
D_{IN} to BL falling [ps]	95.8	137.4	68.1	101.7	94.0
W_{REF} rising to BL rising [ps]	135.5	187.0	100.1	139.3	137.0
W_{REF} rising to BL falling [ps]	207.0	291.8	149.5	209.6	211.0

4.1.4. Sense amplifier

After a read wordline has been activated for a certain read duration, a sense amplifier is needed to retrieve the digital data from the cell using the bitline voltage. For the dynamic cell designs, this is done by comparing the developed bitline voltage to an externally applied reference voltage. For the 2T NW-PR and 3T NW-PR cell designs, the reference voltage will be below half supply, while for the

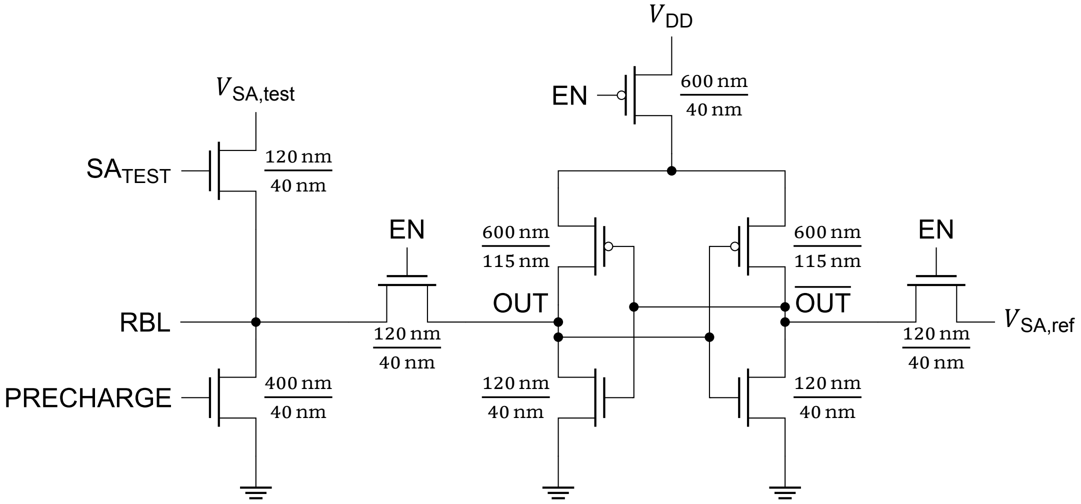


Figure 4.15: Sense amplifier schematic for 2T and 3T NW-PR memory designs.

3T PW-PR cell designs, the reference voltage will be above half supply. Additionally, the 3T PW-PR cell array has double the amount of read bitlines due to the column interleaving. As a result, two sense amplifier designs are used. In the following section, the sense amplifier design for the 2T and 3T NW-PR cell designs is explained. This is followed by the changes needed to arrive at the design for the 3T PW-PR cells.

The sense amplifier has to be small and fast, for efficient layout and low read latency. It must fit in the bitline pitch of the cell array, like the write bitline drivers, and not slow down the read operation significantly. However, since the mismatch between the read bitline voltages generated by the cells is expected to dominate the yield and BER performance, offset and noise requirements are relaxed.

The core of the design is a cross-coupled inverter latch with a power gate and sampling pass transistors as shown in fig. 4.15. It is a simple design using only seven transistors and similar to a design that has been used before in a cryogenic memory [56] but without the additional equalisation transistor. During the sampling phase (EN=1), the pass transistors connect the internal nodes to the bitline and reference voltage while the power gate disables the feedback. During the amplification and latching phases (EN=0), the internal nodes are disconnected from their inputs and the supply is connected through the power gate. With low input voltages, not enough to turn on the NMOS transistors of the inverters, the offset is mainly determined by the threshold-voltage mismatch between the PMOS transistors of the inverters.

Two transistors are added on the bitline side of the sense amplifier: one for precharging the bitline to ground and one for setting an externally applied voltage to the bitline. The precharge is needed to erase data from the previous read operation and ensure that the bitline starts at $0\,\mathrm{V}$ at the start of a read operation. The second transistor allows us to take direct control over both sense amplifier inputs. This can be used to characterise the sense amplifiers in terms of offset and noise by sweeping the two voltages and performing measurements on them (see section 5.3 for details).

The transistors are sized such that the comparison is fast and the offset is not too large without excessive area consumption. All NMOS transistors are minimum size, except for the bitline precharge transistor, which has an increased width of $400\,\mathrm{nm}$. The latch PMOS transistors have a W/L of $600\,\mathrm{nm}/115\,\mathrm{nm}$ to increase their area in order to obtain a sufficiently low offset. The power gate PMOS transistor also has a width of $600\,\mathrm{nm}$, but a minimum length of $40\,\mathrm{nm}$ to minimise its on-resistance. Almost all transistors are implemented using low-threshold devices to ensure that the thresholds of the pass transistors are low enough at cryogenic temperatures and to maximise the speed. Only the NMOS transistors of the inverters are implemented using high-threshold devices, since they could start the comparison if either the bitline or reference voltage is larger than the threshold before the power gate is enabled, resulting in leakage from the reference supply to ground.

The sense amplifier for the 3T PW-PR cell design is similar to the previously described design. The schematic of this sense amplifier design is shown in fig. 4.16. Since the input voltages are high (above half supply), the PMOS and NMOS transistors are swapped. Due to the double bitlines, an additional

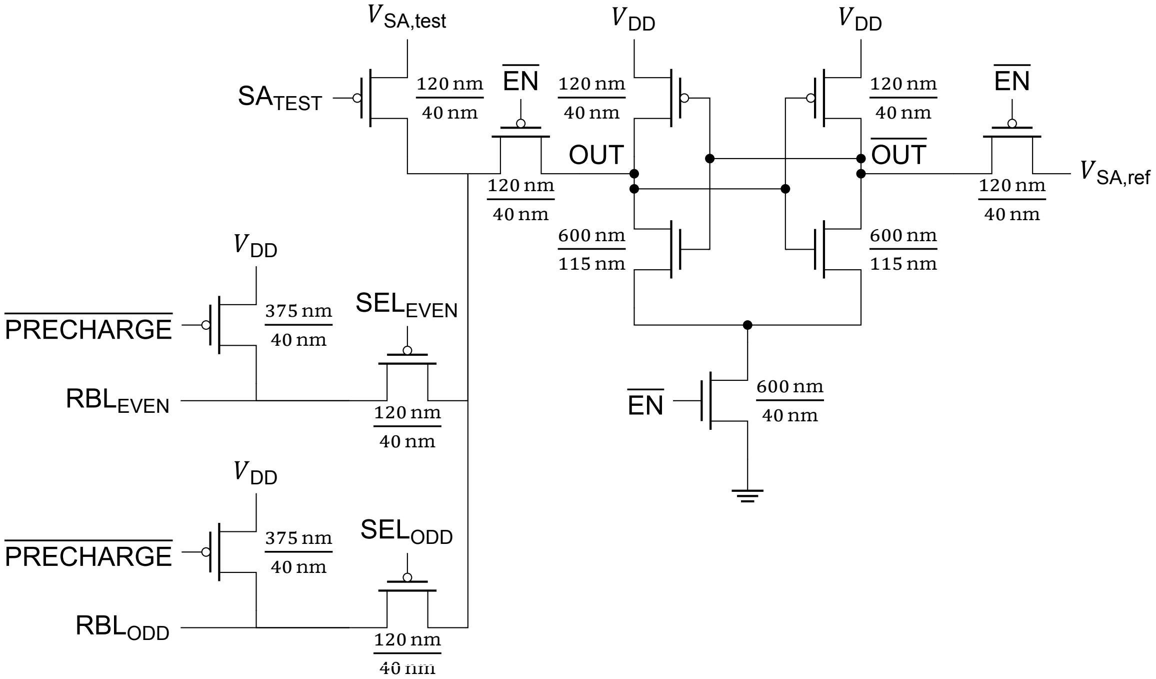


Figure 4.16: Sense amplifier schematic for 3T PW-PR memory design.

set of pass gates is added and each bitline gets its own precharge transistor. Using these pass gates, we can select either of the two bitlines with the least significant bit of the address. This way, only one of the two bitlines has to be discharged during a read operation. If instead the bitlines were connected together, both would have to be discharged, resulting in a doubling of the bitline discharge read energy and read latency.

The layouts of the sense amplifiers can be seen in fig. 4.17. Similar to the write bitline drivers, the designs are similar and stretched to fit the bitline pitch. To improve matching of the transistors that perform the comparison, dummies are added around them. These layouts can all again be stacked to form the full sense amplifier array with well taps and dummies on either side.

The sense amplifiers are simulated to find the decision speed, noise, and offset. Figure 4.18 shows the decision speed of the sense amplifier for the 2T and 3T NW-PR memory designs for input voltages from $190\,\mathrm{mV}$ to $210\,\mathrm{mV}$ with $1\,\mathrm{mV}$ step size and a reference voltage of $200\,\mathrm{mV}$ at $300\,\mathrm{K}$. For small input voltages, the output gives a clear decision within $200\,\mathrm{ps}$ to $250\,\mathrm{ps}$, but with inputs of several 10s of mV to over $100\,\mathrm{mV}$, it will be less than $150\,\mathrm{ps}$. This delay is expected to decrease at $4.2\,\mathrm{K}$.

The input-referred offset and noise are simulated by performing multiple comparisons for fixed, small input voltages, using Monte Carlo and transient noise simulations, respectively. These result in an input-referred offset standard deviation of $12.6\,\mathrm{mV}$ and input-referred noise standard deviation of $3.5\,\mathrm{mV}$. Both are lower than assumed in the memory model in chapter 3, so much better yield and BER performance are expected. Figure 4.19 shows how the input-referred noise standard deviation is computed from the average output of 64 comparisons of a transient noise simulation for several small input voltages. Additionally, it shows an offset of roughly $5.2\,\mathrm{mV}$ due to different loading of the two output nodes, since this simulation uses perfectly matched devices. It is therefore a structural offset for every sense amplifier and can be compensated by decreasing the reference voltage. Mismatch in the load can cause some variation in this offset, which increases the total input-referred offset standard deviation slightly.

4.1.5. Latch

A latch is added to the design to hold the output data of the sense amplifier. It ensures that the read output data is stable during the next read, when the sense amplifiers are in sampling or amplification mode.

The used design is a transmission-gate-based latch using minimum size transistors. The schematic is shown in fig. 4.20 and the associated layout is shown in fig. 4.21. Again, the design is reused for the

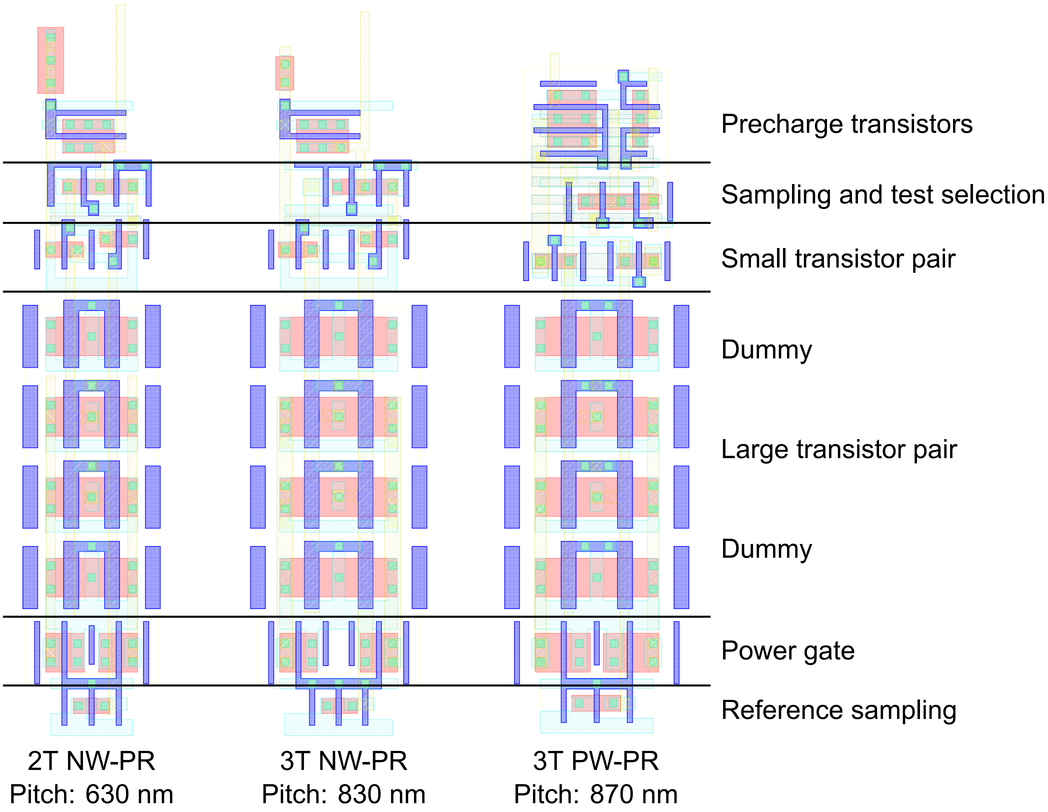


Figure 4.17: Layout of a sense amplifier section for each dynamic cell design. See appendix A for layer legend.

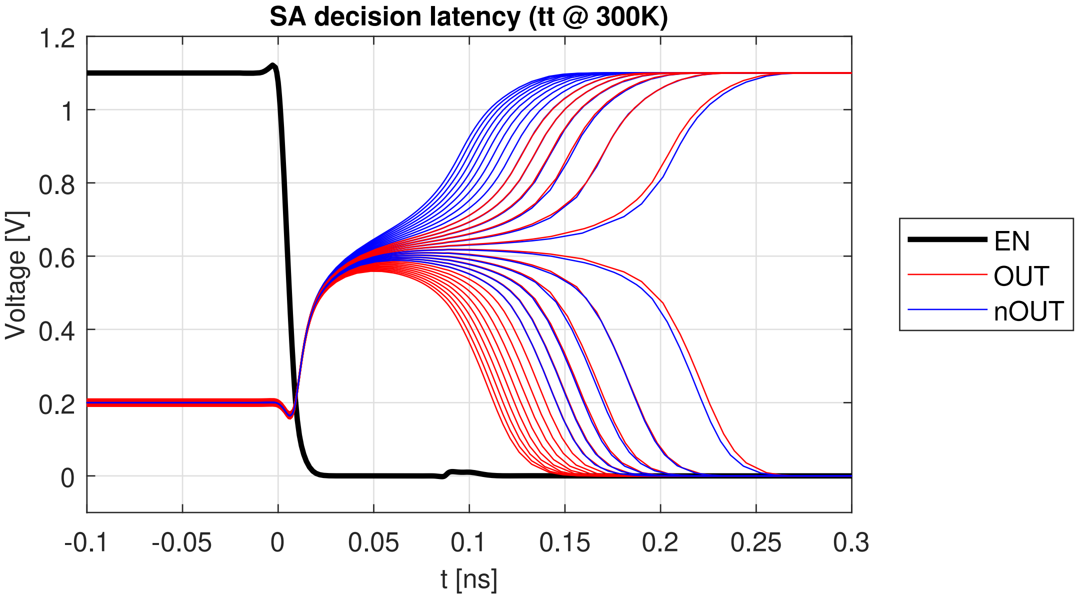


Figure 4.18: Decision simulation of the sense amplifier design for bitline voltages from $190\,\mathrm{mV}$ to $210\,\mathrm{mV}$ with $1\,\mathrm{mV}$ step size and a reference voltage of $200\,\mathrm{mV}$ in the tt process corner at $300\,\mathrm{K}$.

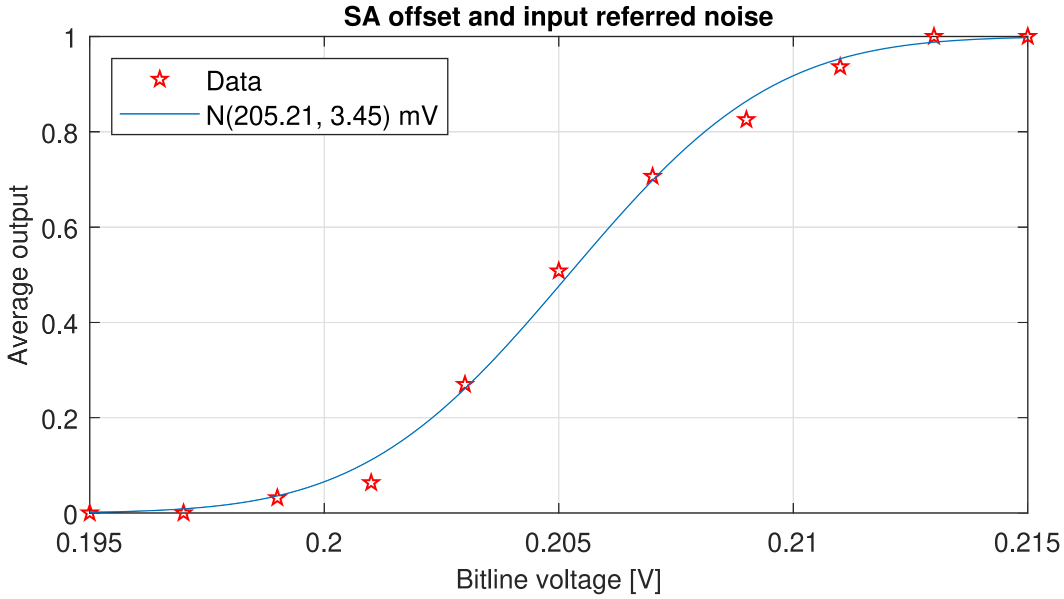


Figure 4.19: Average result of 64 comparisons at various $V_{\text{SA,test}}$ voltages and a reference voltage of $200\,\mathrm{mV}$ on the sense amplifier design for the 2T NW-PR cell memory design. Fitting the data to a normal CDF gives an input referred offset of $5.21\,\mathrm{mV}$ and an input referred noise standard deviation of $3.45\,\mathrm{mV}$.

different cell designs and the layout is stretched to fit the bitline pitch.

The latch is simulated to find the latency and hold time. The clock-to-Q latency of the latch directly adds to the read latency and is approximately $20\,\mathrm{ps}$ to $30\,\mathrm{ps}$. The hold time of the latch is roughly $11\,\mathrm{ps}$ and constrains the minimum delay between making the latches opaque and putting the sense amplifiers in sampling mode, which is determined by the timing generation circuits.

4.1.6. Timing generation

To verify the bitline curves used by the model, we need to be able to vary the duration of the controlling read and write pulses. These controlling pulses can be made using variable width pulse generators by combining these pulses with delayed versions of itself using static logic. This requires a variable, programmable delay chain to create both the variable pulse widths and delays.

The timing generation circuits for the different memories are exactly the same. This reduces the design effort significantly, but also ensures that the timing of the different memories, for the same

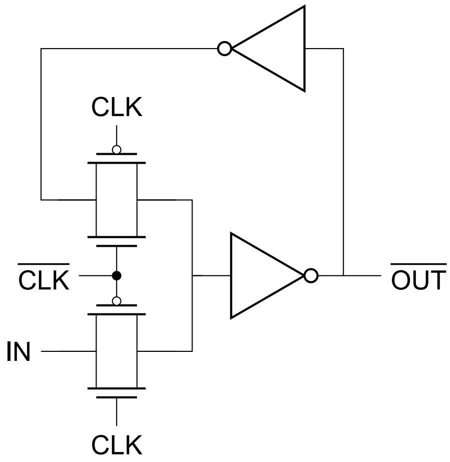


Figure 4.20: Schematic of transmission gate based latch. All transistors are minimum sized.

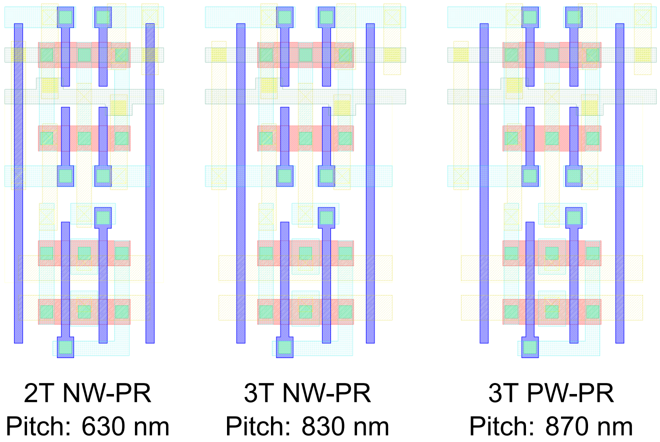


Figure 4.21: Layout of the latches for each dynamic cell design. See appendix A for layer legend.

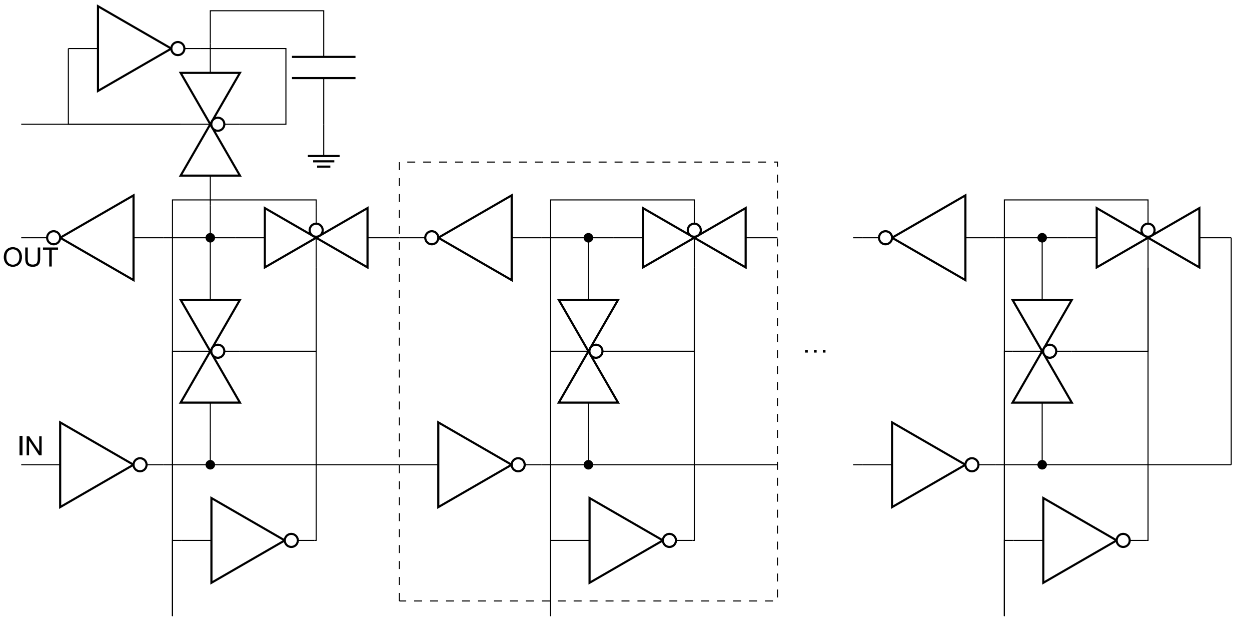


Figure 4.22: Schematic of delay chain constructed of stages with three inverters and two transmission gates (dashed box). Delay settings are entered through the inputs at the bottom to select the chain length (one-hot) and the input at the top left to connect the MOM capacitor.

timing settings, match as good as possible. In some cases, outputs of the timing generation circuits are inverted to obtain the right signal polarity, which introduces one additional gate delay.

Programmable delay chain

A programmable delay chain can be designed by cascading a variable number of inverters. This method is used here, where two inverter chains run in opposite direction. By using transmission gates, the inverter chains are shorted together at a certain point. This point determines the total length of the chain and therefore the total delay. Figure 4.22 shows the schematic of several stages consisting of three inverters and two transmission gates. Stacking multiple stages results in the described situation with two inverter chains running in opposite directions. The number of inverters between input and output can be varied with a multiple of two, resulting in a resolution of roughly $40\,\mathrm{ps}$.

The N- and P-type transistors in the delay stage are sized the same to improve layout regularity and therefore matching between the stages. As a result, all NMOS transistors and PMOS transistors have a W/L ratio of $300\,\mathrm{nm}/40\,\mathrm{nm}$ and $500\,\mathrm{nm}/40\,\mathrm{nm}$, respectively, except for the selection signal inverters shown at the bottom of fig. 4.22. This inverter lies outside the delayed signal path and is not used frequently and therefore uses minimum size transistors to minimise layout area.

To improve the resolution of the delay chain, a metal-oxide-metal (MOM) fringe capacitor is added, together with a transmission gate, at the input of the last inverter. When the transmission gate is enabled, the transition is slowed down by roughly one inverter delay, resulting in a resolution of $20\,\mathrm{ps}$. Since the smallest desired delays are in the order of $150\,\mathrm{ps}$ to $200\,\mathrm{ps}$, this resolution gives an accuracy of at least $13\,\%$ and much better for larger delays.

The layout of a single delay chain stage is shown in fig. 4.23. All transistors are implemented using low-threshold devices to ensure high speed at both room temperature ($20\,\mathrm{ps}$ resolution) and cryogenic temperatures ($< 20\,\mathrm{ps}$ resolution), and the outputs and inputs are located such that multiple stages can be stacked directly to form a longer chain. For very long chains, multiple shorter chains can be stacked vertically. By mirroring each chain across the vertical axis, a meandering chain can be formed which results in a lower aspect ratio at the cost of worse matching due to the mirroring.

Simulations including layout parasitics and Monte-Carlo mismatch show that the resolution of $20\,\mathrm{ps}$ can be achieved. Figure 4.24 shows the delay for each setting of 20 delay chains consisting of 32 stages and the optional capacitor, resulting in 64 possible delay settings. The rising edge delay results in a straighter line than the falling edge, so the rising edge delay is used in the timing generation circuits. The inaccuracy ($\frac{\sigma}{\mu}$) of the rising edge delay due to mismatch is at most $4.5\,\%$ for low settings, but quickly

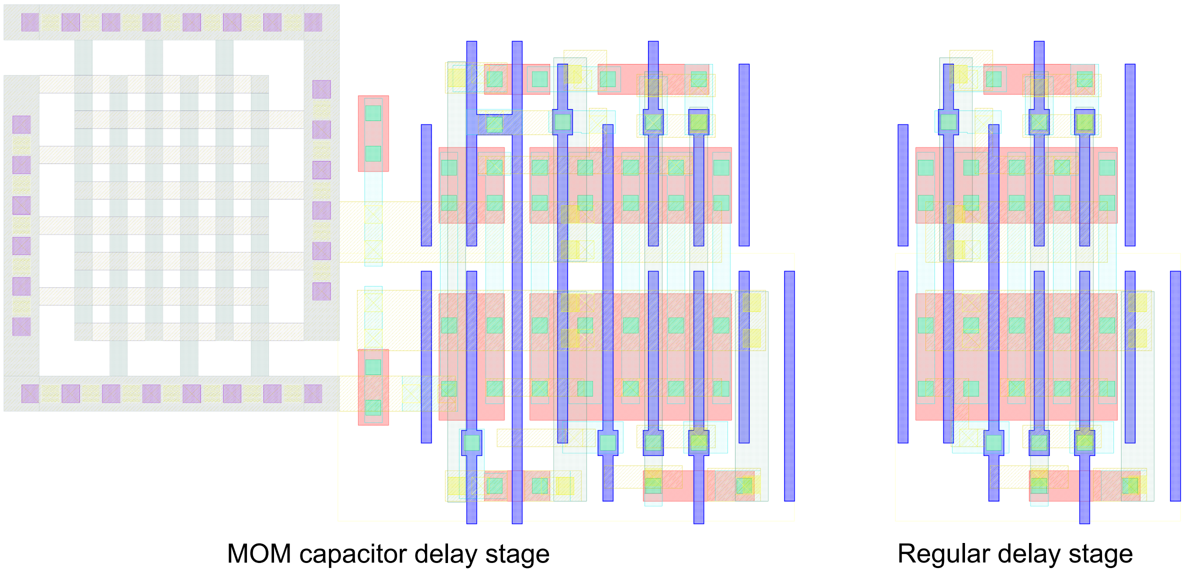


Figure 4.23: Layout of delay chain stages. Left: first stage of a chain, including a MOM capacitor for increased resolution. Right: regular delay stage, used throughout the rest of a chain. See appendix A for layer legend.

reduces to $1\,\%$ or less for settings above 15.

Write timing generation

For the write operations, only a single pulse is needed to enable the write wordline for the desired duration of a write operation. This can be implemented using a pulse generator, followed by driver inverters to drive the entire row decoder.

The schematic of a variable pulse generator is shown in fig. 4.25. On a rising edge at the input, the output will be high for approximately the delay of the delay chain and a single inverter delay. No pulse is generated on the falling edge of the input. Since a write duration of $1\,\mathrm{ns}$ is required to test the model output, we need roughly 50 steps with a resolution of $20\,\mathrm{ps}$. Some margin is taken into account since the delay chain is expected to become faster at cryogenic temperatures, so a delay chain with 64 steps is implemented, using 32 inverter stages and a capacitor. This results in an expected maximum pulse width of almost $1.3\,\mathrm{ns}$.

The layout of the write timing generator is shown in fig. 4.26. The drive strength of the output of the pulse generator is increased using two inverters since it has to drive the clock gating logic gates in the predecoder and the 32 precharge transistors of the row decoder. Using a fan-out of four as guideline, the inverters are sized using $2/1\times$ and $8/4\times$ minimum width for the PMOS/NMOS transistors, respectively.

Simulation results for varying settings show that the desired range of write operation durations can be achieved. Figure 4.27 shows the simulated output pulse width as a function of the delay setting. Fitting a linear expression to this simulation results in the expression shown in eq. (4.2) where N ranges from 1 to 64. The absolute difference between the simulated pulse width and the linearisation is less than $6.5\,\mathrm{ps}$ and less than $3\,\mathrm{ps}$ for settings larger than 4.

$$\text{Pulse width } [\mathrm{ps}] = 19.989 \cdot N + 27.282 \tag{4.2}$$

The latency of the write timing generator is defined as the time between the rising edge of the input and the rising edge of the output pulse and is between $98\,\mathrm{ps}$ and $99\,\mathrm{ps}$ for all settings (typical process corner). This is large enough to ensure that the write bitline drivers have set the bitlines for a refresh operation before the selected wordline is activated by the row decoder.

Read timing generation

For a read operation, a number of pulses are required to properly time different events throughout the memory. These events, the controlling pulses, and the required relations between the pulses are

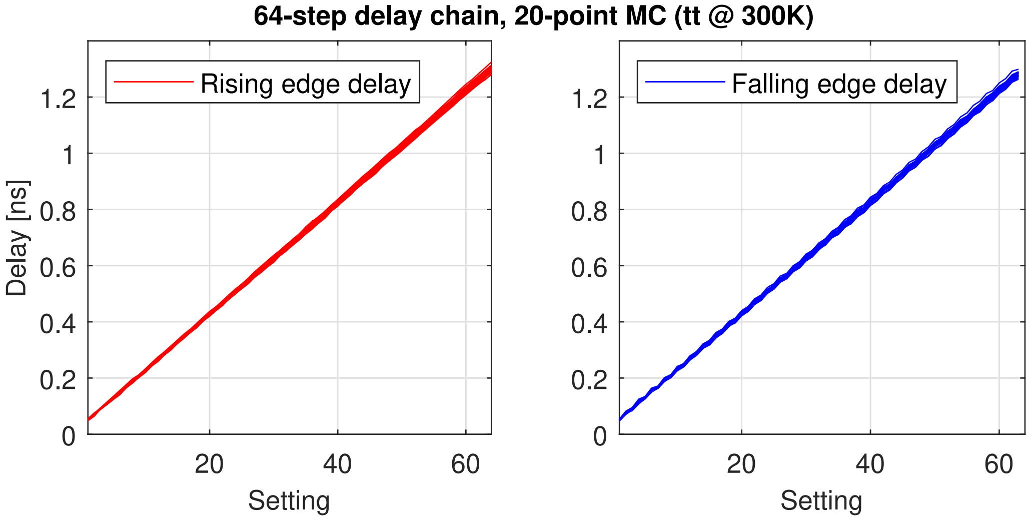


Figure 4.24: 20-point Monte-Carlo simulation of a 64-step delay chain including layout parasitics in the tt corner at $300\,\mathrm{K}$.

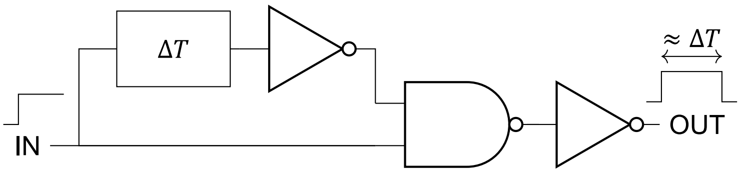


Figure 4.25: General schematic of a rising edge triggered pulse generator, where the ΔT block indicates a delay chain with variable delay.

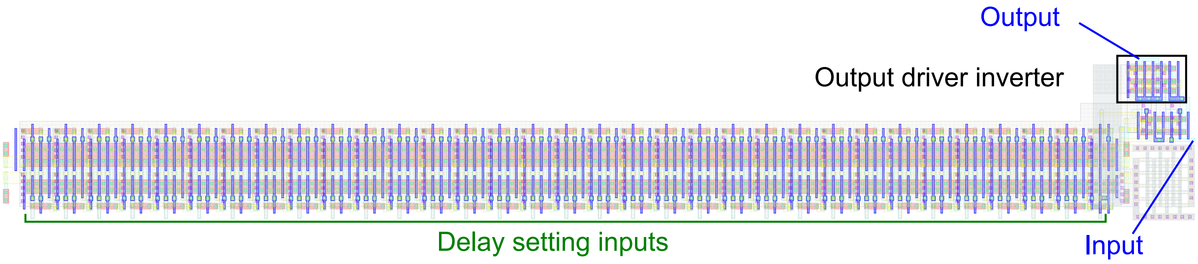


Figure 4.26: Layout of the write timing generator consisting of a 64-step pulse generator and additional output inverters. See appendix A for layer legend.

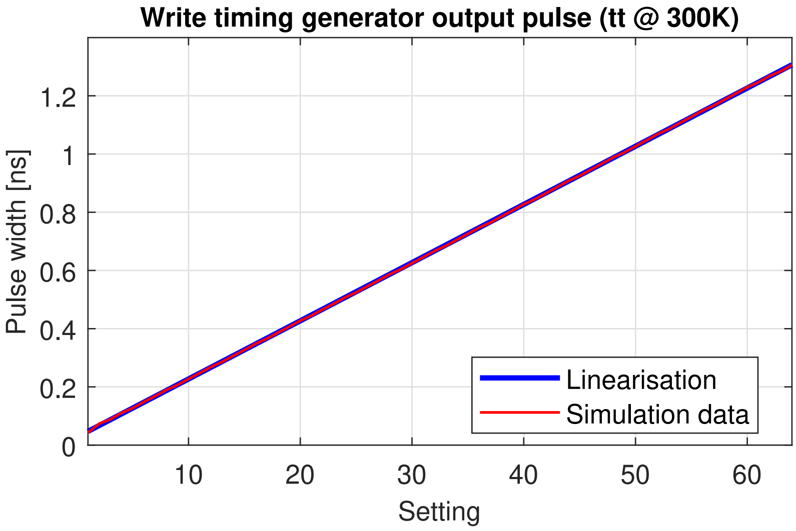


Figure 4.27: Simulated pulse width of write timing circuit as function of the delay chain setting and the best linear fit.

shown in fig. 4.28. First, the latches are turned opaque and the sense amplifier is put in sampling mode. During this phase, the sense amplifier is connected to the precharged bitline, which will also remove the charge from the sense amplifier node. After the sense amplifier has been reset, the bitline precharge is disabled and the read wordline is enabled. This requires non-overlap (NOv) to prevent a short circuit current through selected cells and the bitline precharge transistors, which leads to unnecessary energy consumption. After the bitline voltage has developed, the sense amplifier samples the bitline and reference voltages and starts amplifying the difference, and the read wordline is disabled. Finally, the latches turn transparent again and pass the result of the read operation, and the bitlines are precharged again. A non-overlap is again required to prevent short circuit currents.

Using programmable delay chains, the duration of the sense amplifier precharge and the duration of the read operation can be set. A maximum sense amplifier precharge duration of $300\,\mathrm{ps}$ is desired to guarantee that the sense amplifier is fully cleared, but shorter precharge durations may still result in reliable results. A delay chain with 16 steps can achieve a maximum delay of at least $320\,\mathrm{ps}$. To also investigate the effects of long read durations, a maximum is desired of at least $3\,\mathrm{ns}$. A pulse generator with a delay chain with 192 steps gives a maximum delay of roughly $3.8\,\mathrm{ns}$.

These pulses can be generated with the desired amount of configurability by using a pulse generator with 192 steps, a 16 step delay chain, and some static logic gates. A schematic of the circuit implementing the desired behaviour is shown in fig. 4.29. The sense amplifier test mode signal that enables the transistor connecting the bitline to an externally applied voltage is also included to ensure that the bitline precharge transistors are not connected and the read wordlines are not enabled when the bitlines are forced externally ($\mathrm{SA_{TEST}} = 1$). Large inverters are used on all outputs to ensure that the loads can be driven with fast edges.

Figure 4.30 shows the layout implementing the read timing generation circuit. The pulse generator, delay chain, logic, and output inverters are indicated using coloured boxes.

Simulation results for varying settings show that we can achieve the desired range of precharge and read durations. The total precharge and read duration can be calculated from the settings of the two delay chains with eq. (4.3), where N_{16} and N_{192} indicate the number of steps of the two delay chains that are enabled, ranging from 1 to 16 and 192, respectively. This results in a precharge duration between $114.03\,\mathrm{ps}$ and $405.57\,\mathrm{ps}$, and a read duration between $0\,\mathrm{ns}$ and $3.6\,\mathrm{ns}$, both with a resolution of $20\,\mathrm{ps}$.

$$\begin{aligned} t_{\text{precharge}}\ [\mathrm{ps}] &= 94.59 + 19.44 \times N_{16} \\ t_{\text{read}}\ [\mathrm{ps}] &= 20.01 \times N_{192} - 19.42 \times N_{16} - 183.63 \end{aligned} \tag{4.3}$$

Table 4.3 shows the non-overlap, sense amplifier decision, and input-to-output latency duration of the read timing generator in all process corners at $300\,\mathrm{K}$. The non-overlap durations must be large enough to prevent wasting energy by draining the readout current to the precharge supply. The non-overlap between the falling edge of the read bitline precharge and the rising edge of the read wordline

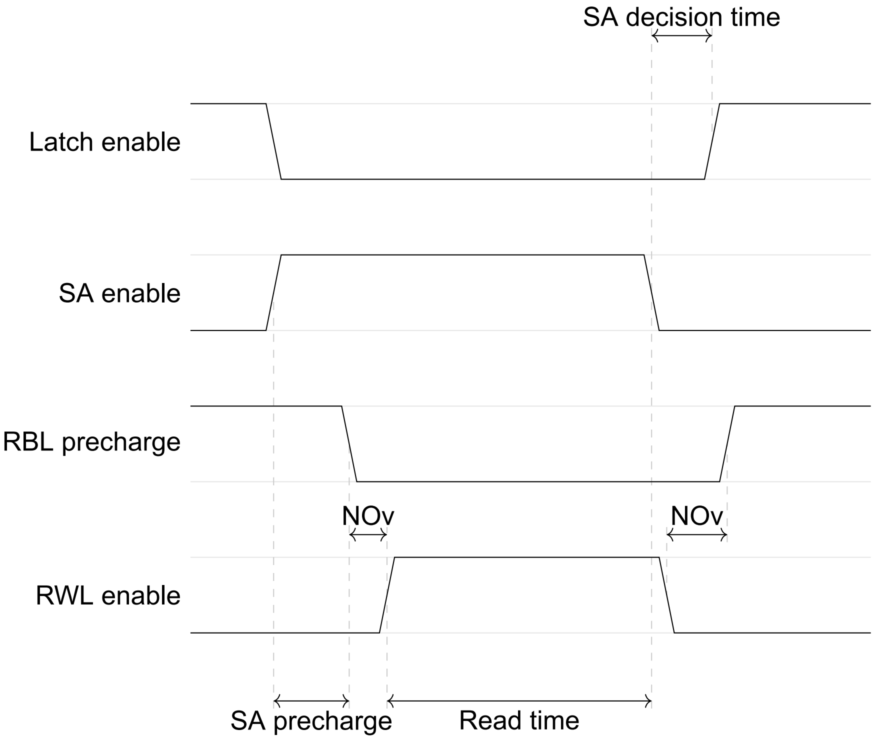


Figure 4.28: Timing diagram showing the required signal timings for a read operation.

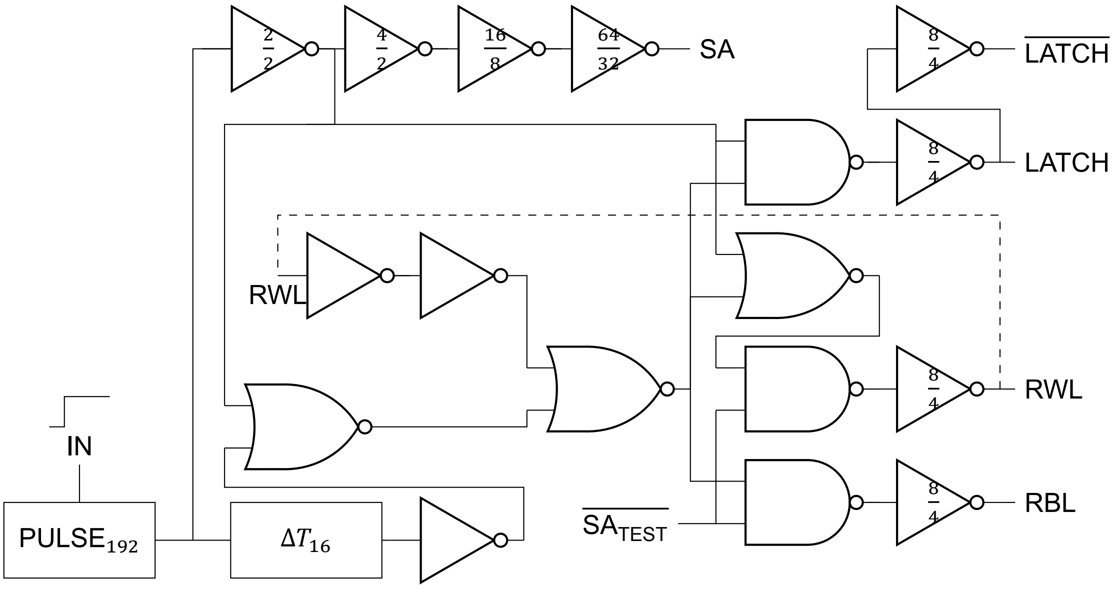


Figure 4.29: Schematic of the read timing generation circuit. The PULSE$_{192}$ block indicates a pulse generator with a 192-step delay chain and the ΔT_{16} block indicates a 16-step delay chain. The ratios in the inverters indicate the PMOS/NMOS transistor width relative to a minimum size transistor. The remaining transistors are minimum width, and all transistors are minimum length.

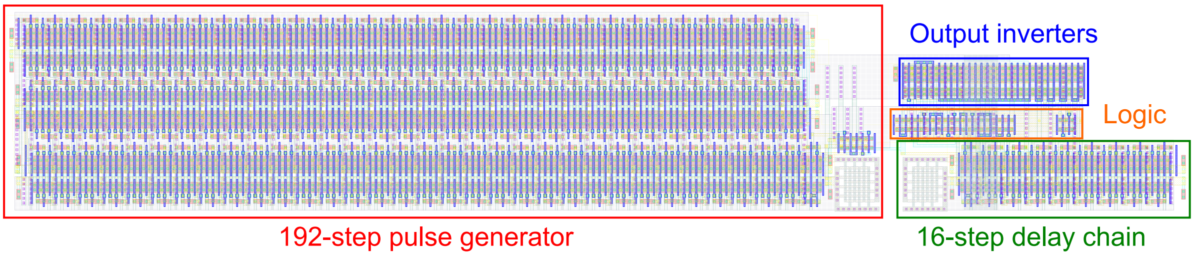


Figure 4.30: Layout of the read timing generator consisting of a 192-step pulse generator, a 16-step delay chain, and additional logic and output inverters. See appendix A for layer legend.

decoder enable signals is $36\,\mathrm{ps}$ to $81\,\mathrm{ps}$ across process corners. With the additional rising edge latency of the row decoder, this guarantees more than $100\,\mathrm{ps}$ non-overlap between the wordline voltage pulse and precharge transistor gate voltage pulse in every process corner. The non-overlap between the falling edge of the read wordline decoder enable and the rising edge of the read bitline precharge signals is $138\,\mathrm{ps}$ to $281\,\mathrm{ps}$ across process corners. This is $45\,\mathrm{ps}$ to $120\,\mathrm{ps}$ longer than the falling edge latency of the row decoder in the same process corner. This ensures that there is no short circuit through the cells.

The sense amplifier decision time must be large enough to prevent latch kickback interference. The sense amplifier decision time is $167\,\mathrm{ps}$ to $337\,\mathrm{ps}$, which gives enough time for most inputs (see fig. 4.18). Only for small inputs that get close to metastability, the kickback of the latch will interfere with the decision. However, increasing the decision time will slow down the read operation latency. Including the trigger to sense amplifier enable delay, around $450\,\mathrm{ps}$ (tt corner) of latency overhead needs to be added on top of the duration of the sense amplifier precharge and read operation for the total read operation latency.

Table 4.3: Read timing generation non-overlap and sense amplifier decision time across corners.

Process corner	tt	ss	ff	fs	sf
RBL precharge to RWL enable [ps]	53	81	36	51	56
RWL enable to RBL precharge [ps]	193	281	138	200	189
Sense amplifier decision time [ps]	233	337	167	241	228
Trigger to sense amplifier enable [ps]	156	223	113	148	163

4.1.7. Full memories

The previously mentioned circuits can be combined to create a full memory, according to the schematic shown in fig. 4.31. The inverters shown in cyan need to be added for the 3T PW-PR memory to ensure that the timing signal polarities are correct. Since pitch matching between the peripherals and cell core has been taken into account from the beginning, the layout is compact and regular.

The layouts of the three dynamic memories can be seen in fig. 4.32. Two versions of each memory are put on chip: one with the described cell arrays, and one with cell arrays where the high- and standard-threshold devices have been replaced with standard-/low-threshold devices, respectively. Additionally, the final arrays are covered by a power and ground grid, created using the fourth and fifth metal layer for power distribution. These layers are not visible in the figure in order to make the different memory components visible.

The timing circuits are responsible for a large part of the area and power consumption due to the high number of switching nodes. In an actual memory use-case, these would be replaced with less flexible timing circuits based on, for example, dummy lines and cells in the array. This saves a lot of area and power and still ensures that enough time has passed to generate enough signal on the bitlines. To ensure that the timing circuits do not interfere with the power consumption measurements on the core itself, separate supplies are used.

Simulations of the full memories show that all memory operations work. Data can be written, refreshed, and read successfully and with various timing settings.

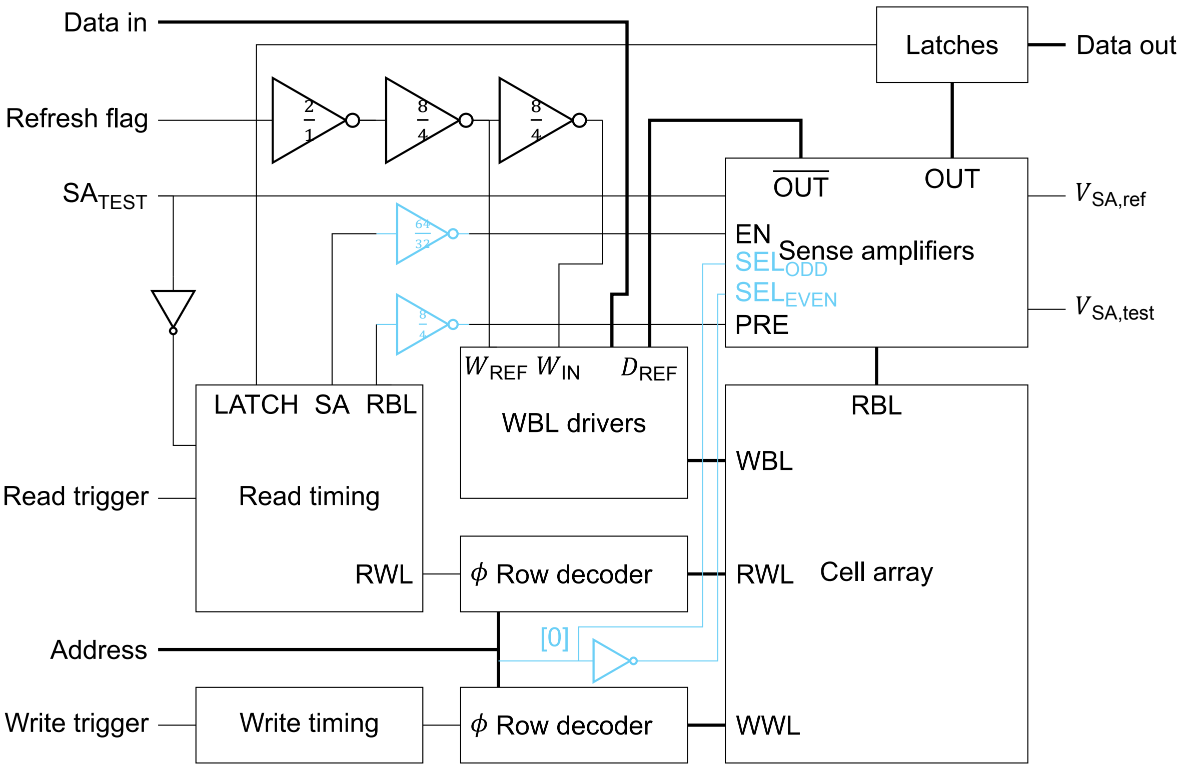


Figure 4.31: Schematic of a dynamic memory using all previously mentioned blocks and some additional inverters. The inverters and interconnect in cyan are only needed for the 3T PW-PR design.

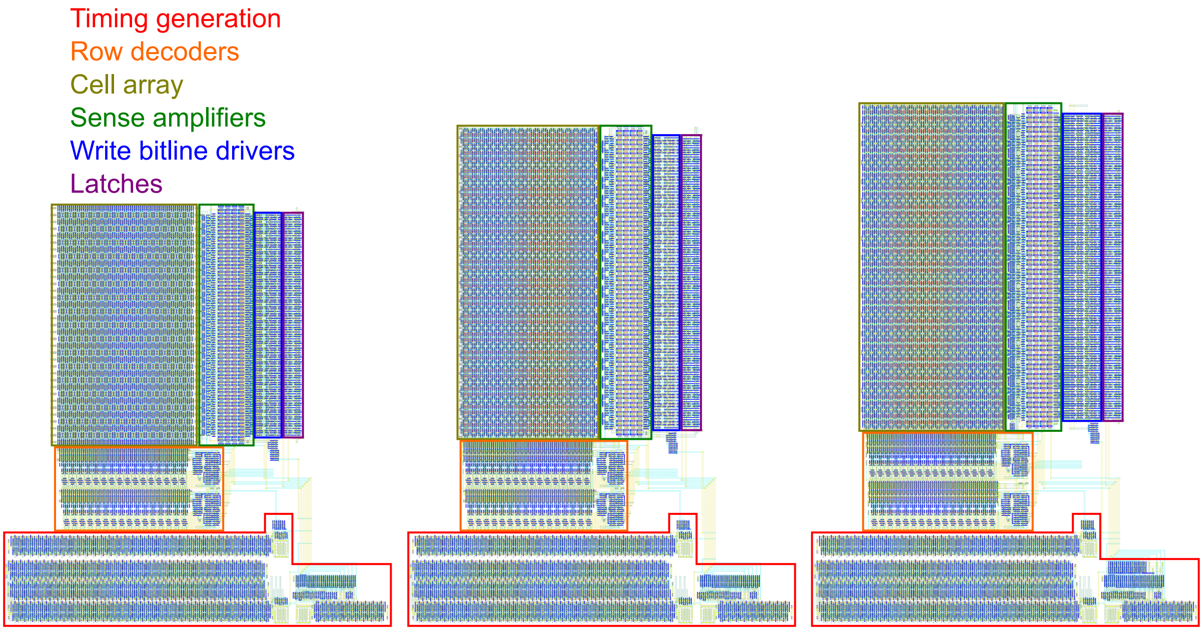


Figure 4.32: Layout of the three dynamic memory designs. The coloured boxes indicate the different memory components. See appendix A for layer legend.

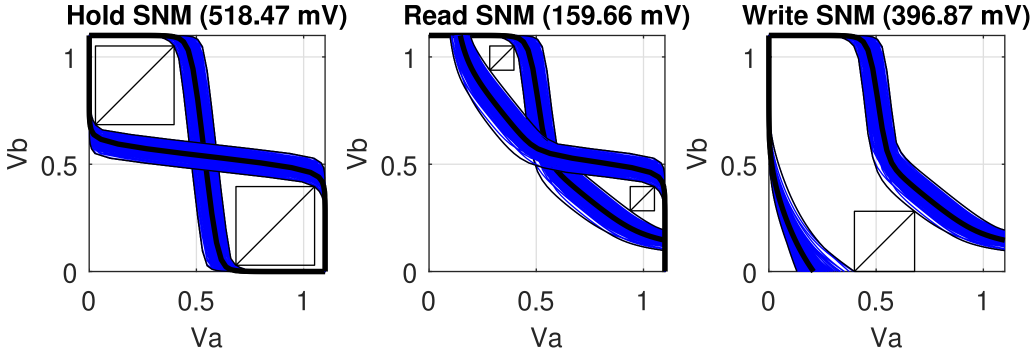


Figure 4.33: Static noise margin simulations of a 6T static cell with $\beta = 1.5$ and all standard threshold devices at $233\,\mathrm{K}$, using a 300-point Monte-Carlo simulation with process (global) and mismatch (local) variation.

4.2. Static memory

The design of the static memory is similar to the dynamic memories, but there are some significant differences. The static cell uses a single wordline, which means that only a single row decoder is needed which is shared between write and read instructions. Additionally, the cells still use two bitlines, but they are used differentially. This single pair of bitlines is also used for both read and write operations, which requires additional coordination.

In this section, we will first look at the design of the static cell array. This is followed by the design of the peripherals, specifically the bitline driver and sense amplifier. The timing generation is shown next, including the additional coordination necessary for the static cell design. The cell array and peripherals are then combined to form a full memory. Finally, an additional structure for measuring the SNM of the cell design is shown.

4.2.1. 6T cell array

We start the static memory design by looking at the static cell array. The 6T static cell is the most used static cell due to its small size, regular layout pattern, and proven design in many applications. It consists of a bi-stable circuit consisting of two cross-coupled inverters and two pass transistors between the internal nodes and the differential bitline pair. The schematic and further operation details of this cell type can be found in section 2.1.2.

Designing a small and optimal 6T static memory cell array is not simple and often requires pushing technology nodes to their limits to get the highest density. Design rules prohibit us from achieving industry-standard high densities, but this gives a fair comparison baseline for the dynamic cells which are designed using the same rules.

The schematic of the 6T static memory cell is shown in fig. 2.7. The load transistors (PL/PL') are minimum size transistors since they are only there to supply leakage current to the high-side node and ensure that the data is not lost. The access transistors (NA/NA') are also minimum size to ensure that a read operation does not cause the cell to flip. The driver transistors (ND/ND') also have a minimum length, but a width equal to $1.5\times$ minimum width. This ratio between driver strength and access strength is called β and is in this case therefore 1.5. Increasing the β will make the cell easier to read, but harder to write. Typically, a β of at least 1.5 is used [21]. This leads to a stable cell, shown by the Monte Carlo SNM simulation figures shown in fig. 4.33 under both process (global) and mismatch (local) variation, with a worst case read SNM of $159.66\,\mathrm{mV}$ for 300 half-cells.

Different cell layouts have been proposed over time, but for modern technology nodes ($<90\,\mathrm{nm}$) the *Lithographically Symmetrical* (LS) layout is often used [21]. On the left side of fig. 4.34, the tileable LS layout of the designed memory cell is shown (distorted to protect exact dimensions). Similar to the dynamic cells, the bitlines again run horizontally on the second metal layer along the smaller cell dimension direction, while the wordlines run vertically on the third metal layer along the larger cell dimension direction. The effective cell dimensions are $1.28\,\mathrm{\mu m}$ by $0.42\,\mathrm{\mu m}$, which equals a cell area of $0.5376\,\mathrm{\mu m}^2$.

This cell layout can be tiled into an array with a ring of dummies and two columns of well taps. The resulting layout is shown on the right side of fig. 4.34. The total cell array area is approximately

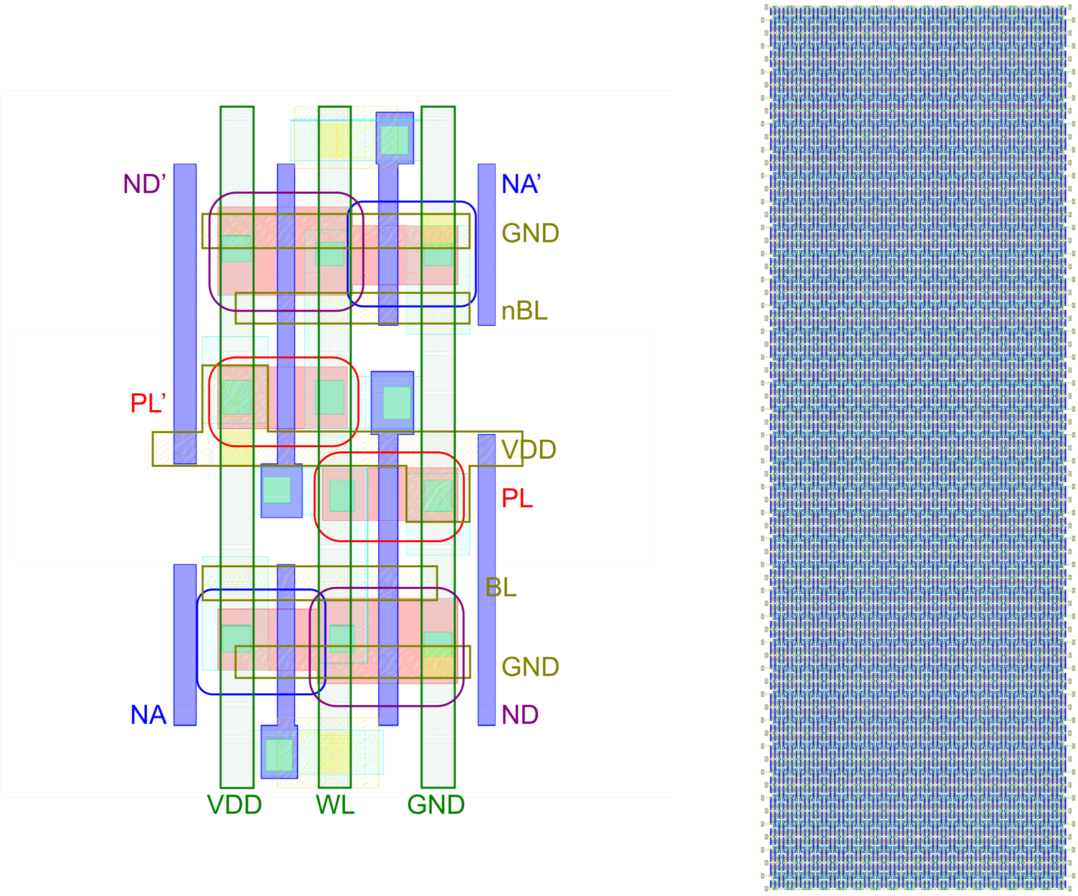


Figure 4.34: Left: (distorted) LS layout of a 6T static memory cell with highlighted inverter pair (PL/PL’ and ND/ND’), access transistors (NA/NA’), wordline (WL), complementary bitlines (BL/nBL), supply (VDD), and ground (GND). Right: layout of 6T static memory cell array with 32 rows, 32 columns, dummies, and well taps. See appendix A for layer legend.

683 μm^2. Similar to the dynamic memories, one array is made using standard-threshold devices and one array is made using low-threshold devices to compare room temperature and cryogenic temperature performance.

4.2.2. Bitline driver

Due to the use of a single complementary bitline pair that is used for both read and write operations, the bitline driver has to become more complex. The memory can be in three operating modes: write, read, and hold. In case of the write mode, the bitlines have to be driven differentially based on input data. During the read mode, both bitlines need to be floating and the cells that are being read will develop a voltage. When in hold mode, the bitlines have to be precharged to the supply to ensure that a read operation can start at any moment.

The output stage of the bitline driver consists of an NMOS and PMOS transistor. From the operating modes, we can derive the desired gate voltage levels for the two transistors in each mode. Table 4.4 shows a truth table, indicating the gate voltages needed for the bitline driving transistors as a function of the mode and data input. A similar table holds for the transistors driving the complementary bitline, except the data input is inverted.

Table 4.4: Truth table for gate voltages of bitline driving transistors for different modes and data.

Mode	Data	Description	PMOS gate voltage	NMOS gate voltage
Write	0	Pull bitline down	1	1
Write	1	Pull bitline up	0	0
Read	-	Leave bitline floating	1	0
Hold	-	Pull bitline up	0	0

The mode is derived from outputs of the timing circuitry. The read mode is recognised based on an RBL signal which is low when the bitlines must be floating. The write mode is recognised based on a WBL signal which is high when the bitlines need to be driven differentially. When neither conditions are met, the bitlines must therefore be precharged to the supply. Note that both conditions can not be met at the same time, as this would imply a simultaneous write and read operation.

Since the logic has to be done in a small area, within the width of the cell bitline pitch, a slightly unconventional logic style is used where both gate voltages are created within a single logic gate. The schematic of the bitline driver, including the driver transistors, is shown in fig. 4.35. To ensure fast charging and discharging of the bitline, the driver transistors are larger than minimum size. Additionally, to speed up the (dis)charging of their gates, the logic PMOS transistor widths have been increased. This design is smaller than a fully static design and ensures non-overlap between the gate voltages since the PMOS gate voltage is always larger than or equal to the NMOS gate voltage, which means that a short circuit through the driver transistors is not possible. Note that there is a potential short circuit path in the logic when RBL is low, WBL is high, and DATA is high, but this requires a simultaneous read and write operation which can never be applied by the hardware driving the memory (see section 5.1.3).

Figure 4.36 shows the layout of the bitline driver for the static cell design, which fits in the bitline pitch of the cell layout. The coloured boxes indicate the logic, the driving transistors for both bitlines, and an inverter for the data for the inverse bitline. Similar to the layouts shown before, contact vias are shared along the edges to improve density. This allows the design to be stacked into an array. Since the array gets long ($> 40\,\mu m$), placing well taps at both ends is no longer sufficient. Therefore, well taps are added throughout the array to ensure a well-defined substrate potential, even at cryogenic temperatures. To improve speed at room temperature, all transistors are implemented using low-threshold devices. At cryogenic temperatures, the speed is expected to be even higher.

4.2.3. Sense amplifier

The core of the sense amplifier design is copied directly from the 3T PW-PR dynamic memory. During readout, the bitlines for the static cell design also start precharged to the supply, and get discharged by the cell. This requires the power-gated NMOS pair also used in the 3T PW-PR memory sense amplifier. Copying the core of the design also reduces design effort and again allows for a fair comparison with the same peripherals. Only the length of the main NMOS pair transistors is increased from 115 nm to

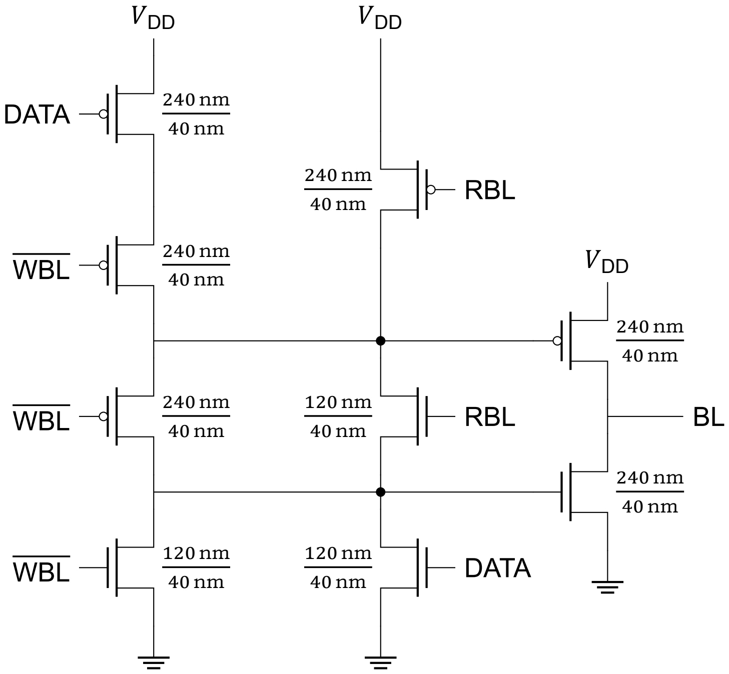


Figure 4.35: Schematic of the bitline driver for the 6T static cell memory design. When RBL is low, BL must be floating. When WBL is high, DATA must be copied to BL. When RBL is high and WBL is low, BL must be pulled up.

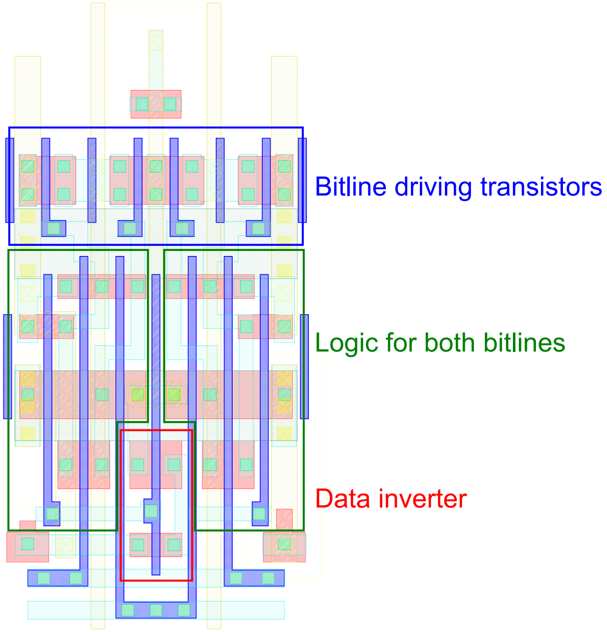


Figure 4.36: Layout of the complementary bitline driver for the 6T static cell memory design. See appendix A for layer legend.

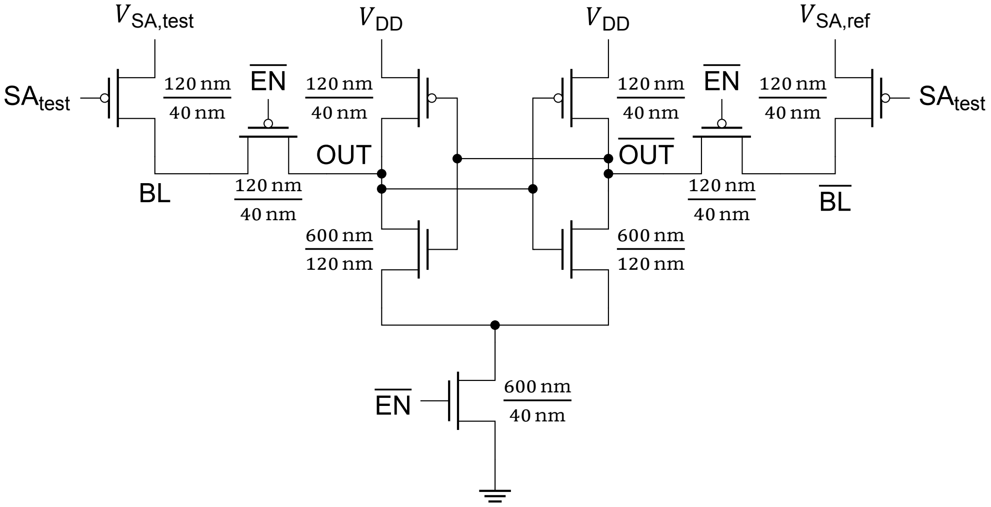

$V_{\text{SA,test}}$
V_{DD}
V_{DD}
$V_{\text{SA,ref}}$
SA_{test}
$\frac{120\,\text{nm}}{40\,\text{nm}}$
$\overline{\text{EN}}$
$\frac{120\,\text{nm}}{40\,\text{nm}}$
$\frac{120\,\text{nm}}{40\,\text{nm}}$
EN
$\frac{120\,\text{nm}}{40\,\text{nm}}$
SA_{test}
OUT
$\overline{\text{OUT}}$
BL
$\frac{120\,\text{nm}}{40\,\text{nm}}$
$\frac{120\,\text{nm}}{40\,\text{nm}}$
$\overline{\text{BL}}$
$\frac{600\,\text{nm}}{120\,\text{nm}}$
$\frac{600\,\text{nm}}{120\,\text{nm}}$
$\overline{\text{EN}}$
$\frac{600\,\text{nm}}{40\,\text{nm}}$

Figure 4.37: Schematic of the sense amplifier for the 6T static cell memory design.

$120\,\text{nm}$ for layout regularity reasons.

The difference with the dynamic memories is that both inputs of the sense amplifier are now connected to the array. Instead of comparing a bitline against a reference, the complementary bitlines are compared against each other to determine which is the high side and which is the low side.

To allow for characterisation of the sense amplifiers, an additional transistor is placed on either side. This leads to the sense amplifier schematic shown in fig. 4.37. Note that there is no need to include the bitline precharge transistors since the bitline drivers already ensure that the bitlines are precharged. The latch core and access transistors are sized identical to the 3T PW-PR memory sense amplifier design. The characterisation pass transistors are fast enough at minimum size, and all transistors are implemented using low-threshold devices, except for the latch PMOS transistors which are again implemented using high-threshold devices to prevent leakage.

The layout of the sense amplifiers is shown in fig. 4.38. It is similar to the layout of the sense amplifiers for the dynamic cell arrays to match performance again. A ring of dummies is included around the NMOS transistors that should be matched. Due to the much larger bitline pitch than the dynamic cells, additional dummies are placed between neighbouring sense amplifiers to ensure a regular layout.

4.2.4. Row decoder and latch

The row decoder and latch designs are reused from the dynamic memories. Since the wordline and bitline pitches are different, both layouts are stretched to match the new pitches.

4.2.5. Timing generation

The timing generation circuits from the dynamic memories are reused for the static memory to ensure that we get similar pulse shapes for the same settings. However, due to the shared nature of the wordlines and bitlines for read and write operations, the timing pulses need to be combined.

The row decoder enabling signal can be created by combining the wordline enabling signals from both timing generation blocks through an OR-gate. To ensure that write operations can not be executed when the sense amplifiers are characterised ($\text{SA}_{\text{test}} = 1$), the wordline enabling signal from the write timing circuit is gated by the sense amplifier characterisation signal.

The bitline signals are kept separate going into the bitline driver, since the actions are different for read and write operations. The bitline signal for the read operation is already generated by the read timing circuit and can directly be connected. To generate the bitline signal for the write operation, the write wordline signal is delayed using a delay chain and the resulting signal is combined with the original wordline signal using an OR-gate. This ensures that the bitline is driven at least until the end of the wordline enable pulse, with programmable extension to cover the row decoder delay. The resulting signal is again gated by the sense amplifier characterisation signal to prevent driving the bitlines when they are connected to the external reference voltages.

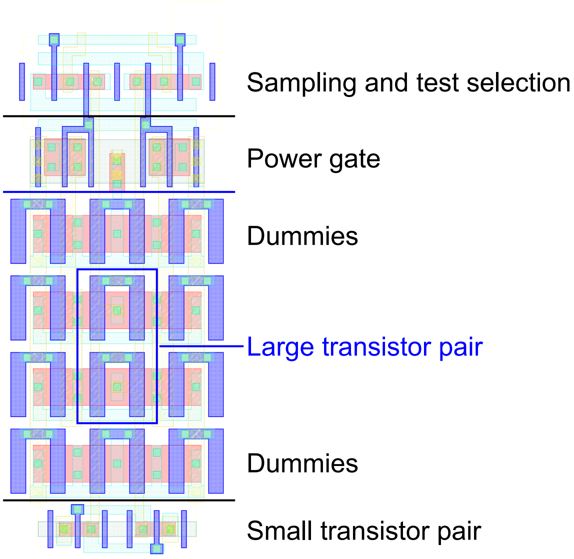


Figure 4.38: Layout of the sense amplifier for the 6T static cell memory design. See appendix A for layer legend.

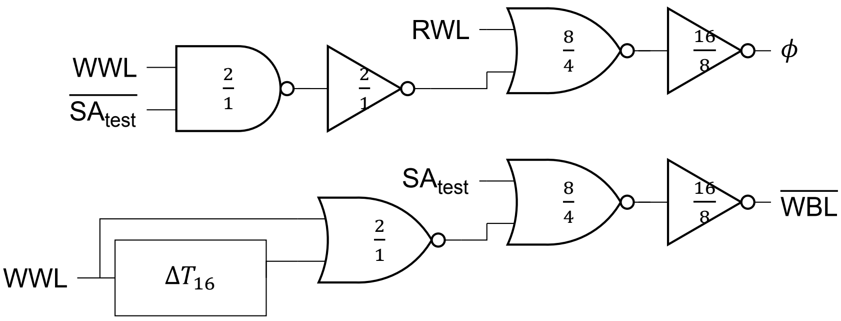


Figure 4.39: Schematic of the timing pulse combining logic for the 6T static cell memory design.

The schematic of the pulse combining logic is shown in fig. 4.39. The logic gate sizes are shown for PMOS/NMOS and indicate their width as a multiple of minimum width, with all lengths equal to minimum length.

4.2.6. 6T cell memory

The previous circuits can be combined to form the full memory, which results in a layout similar to the dynamic memories. The resulting layout is shown in fig. 4.40. The coloured boxes indicate the different parts of the memory, and again it will be covered by a power and ground grid on the fourth and fifth metal layers for power distribution.

4.2.7. SNM array

As mentioned before, a commonly used tool in static memory cell design is the *Static Noise Margin* (SNM). To verify changes in the SNM at cryogenic temperatures, it has to be directly measured.

To accurately measure the SNM of a cell design, the measurement structure should resemble an array. This ensures that the environment, and therefore stresses and proximity effects, are also encountered in the measurement structure. As a result, the measured characteristics are expected to match the characteristics of actual memory cells in an array. Additionally, the characteristics of cells will vary due to mismatch, so using an arrayed measurement structure allows for measuring multiple cells. The data can then be used to derive cell statistics which are important due to the large number of cells typically used in a memory array.

The arrayed SNM measurement structure is based on a structure proposed in literature [72]. To

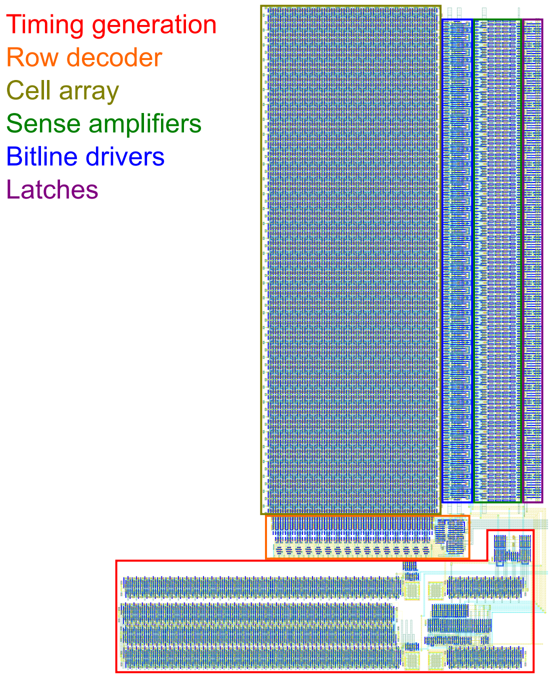


Figure 4.40: Layout of the full static memory design. The coloured boxes indicate the different memory components. See appendix A for layer legend.

simplify the structure slightly, the input transmission gate has been removed and all inputs are connected directly. Additionally, only two levels of output transmission gates are used. The first layer selects a cell from each array column by using the wordlines, and connects them to the output bitlines. A second layer selects a column and connects its bitline to the output. The transmission gate hierarchy is shown in fig. 4.41 for a 4×4 array. Tri-state drivers at the top of the figure are used to drive the input bitlines high or low for measuring the read and write SNMs. Leaving the bitline floating results in the same effect as keeping the wordline low, since there is no DC current flowing through the access transistor, and is used to measure the hold SNM without the need for an additional wordline. An additional second-layer transmission gate, in the bottom right of the figure, gives the possibility to connect the input directly to the output for characterisation of the source follower which is treated later. To be able to characterise cells made from both standard-threshold and low-threshold devices, half the columns use standard-threshold cells, while the other half uses low-threshold cells.

To achieve the array-like environment, a group of four cells is altered in such a way that two cells are replaced with selection transmission gates without breaking the array regularity. Figure 4.42 shows the layout of such an array section and a section of the actual static cell array (distorted to protect exact dimensions). The measured inverters and pass transistors are shown in the blue boxes. The layout environment is similar up to about $1\,\mu\mathrm{m}$ away as indicated by the green boxes. Between these measured structures, two minimum size transmission gates and an inverter are located, which implement the first selection layer. The complete array with dummies around the edge, well taps, wordline and bitline drivers, and the second layer of transmission gates is shown in fig. 4.43.

The impedance of minimum size transmission gates for mid-rail voltages can increase significantly at cryogenic temperatures due to the increase in threshold voltage and mismatch. To prevent the impedance getting to large, the transmission gates are implemented using low-threshold devices. Additionally, the bulk terminal of the PMOS transistors of the transmission gates is exposed to a chip pad which allows for back-biasing to further reduce the threshold voltage. If the impedance still increases excessively, leakage at the output of the chip and in the measurement setup can cause the output voltage to drift, preventing accurate measurement.

Source follower

To ensure low output impedance from the chip, the output signal is buffered by a source follower. The schematic and layout of the source follower are shown in fig. 4.44. The source follower is implemented

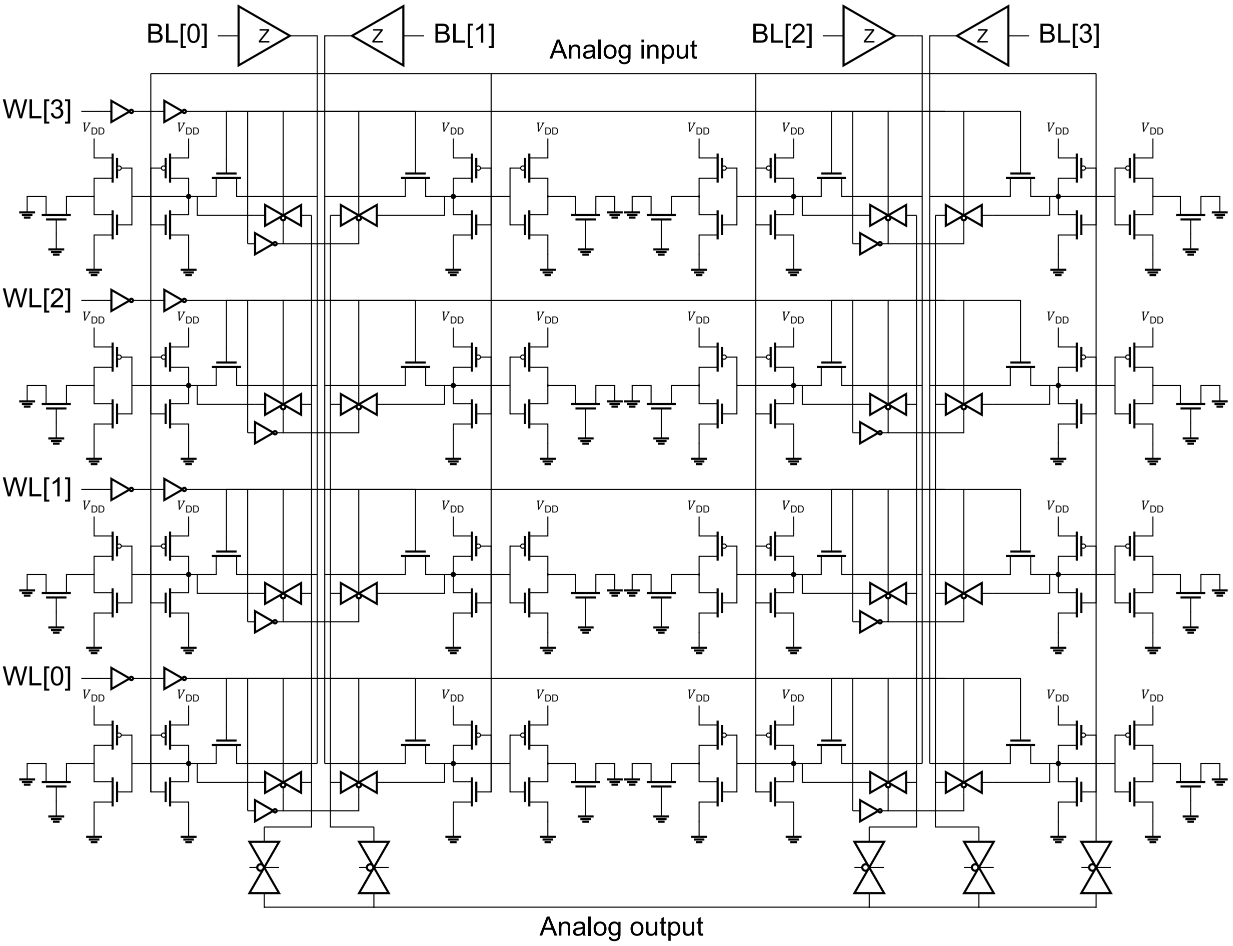


Figure 4.41: Schematic of a part of the SNM measurement array (4×4). The active wordline is supplied externally from a static decoder. The bitlines are driven using tri-state buffers (indicated by the Z), and the column for the output is also selected externally (inverters for the transmission gates not shown). Using the right-most column select transmission gate, the analog input can be directly connected to the analog output.

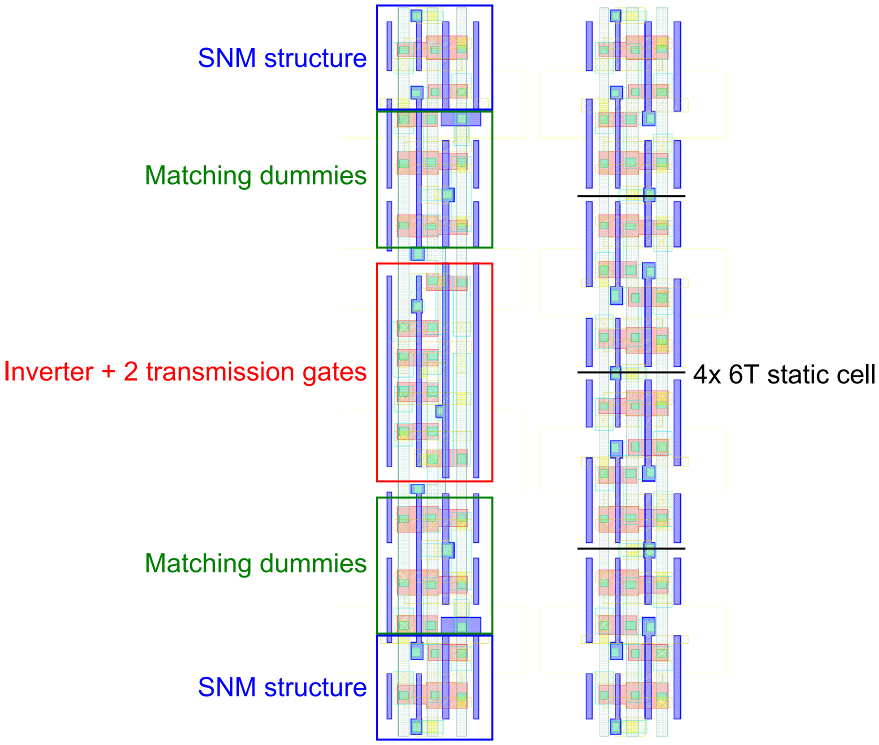


Figure 4.42: (Distorted) layout of the SNM measurement array section. The measured SNM structures are indicated by the blue boxes. The transistors in the green boxes act as dummies to improve matching with an actual static cell array. The red section indicates the selection transmission gates and inverter for the transmission gates.

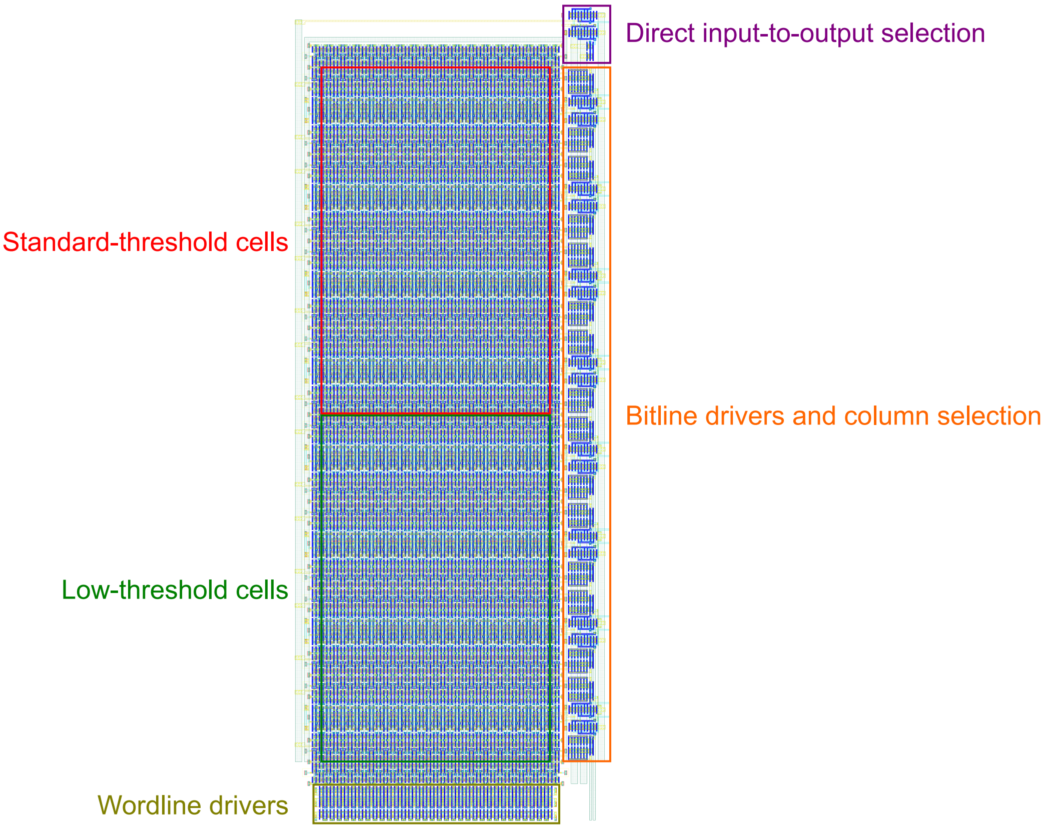
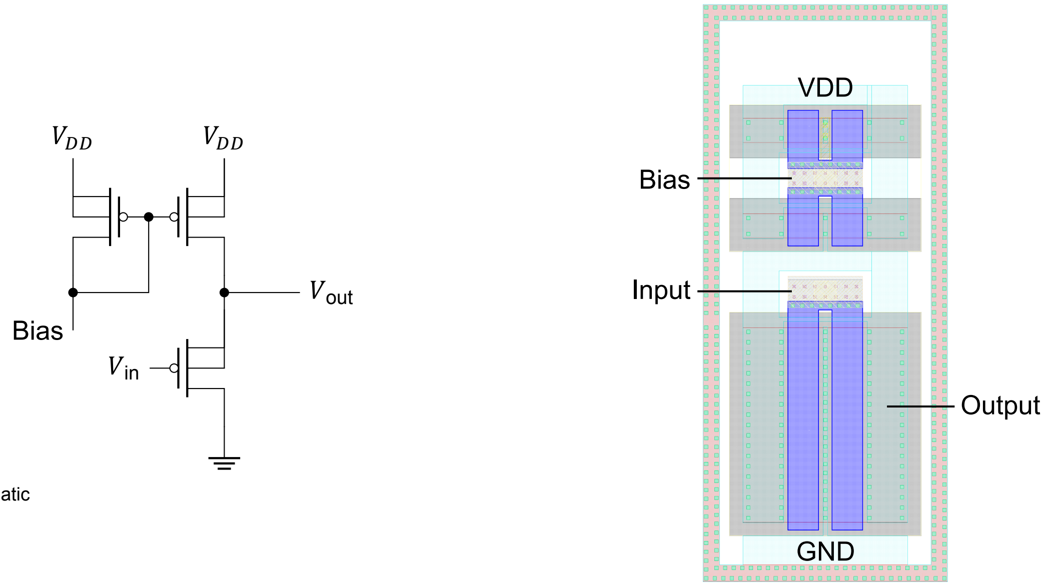

Direct input-to-output selection
Standard-threshold cells
Bitline drivers and column selection
Low-threshold cells
Wordline drivers

Figure 4.43: Layout of the full SNM measurement array.



(a) Schematic

(b) Layout (see appendix A for layer legend)

Figure 4.44: Schematic and layout of the output source follower for the SNM measurement array.

using thick oxide devices and operated with a supply of $2.5\,\mathrm{V}$ to ensure enough headroom for the threshold voltage shift between input and output.

The biasing current mirror allows for three possible biasing schemes. In the first case, current biasing through the current mirror can be used by pulling a current from the bias node. Second, the source follower bias transistor can be voltage biased directly by applying a voltage to the bias node. Finally, the current mirror can also be disabled by pulling the bias node to the supply voltage. In that case, the input transistor can be current biased through the output by pushing a current into the output node.

The mapping between input and output can be calibrated since the array contains an additional second-level transmission gate connecting the input directly to the output. By sweeping the input voltage and measuring the output voltage, the transfer characteristic of the source follower can be measured. This can be used to find the input voltage of the source follower from one of the cells by measuring the source follower output.

4.3. Conclusion

In this chapter, several full memories have been designed with different cell designs. The peripheral designs are varied as little as possible to ensure that any differences in performance are due to cell design. Programmable delay chains are used to generate the timing signals which allows the cells to be characterised with various signal timings. For the static cell design, an SNM measurement structure is added such that the SNM of the design can be measured at both room temperature and $4.2\,\mathrm{K}$.

5 Design of test chip and test plan

In the previous chapter, we have designed several memories to be tested and characterised. Due to the large number of digital inputs and outputs of the memory cores and the limited number of chip pads, additional testing hardware is needed. It must be able to apply various operation sequences to all memories at high speeds and reduce the amount of pads needed.

In this chapter, we will first cover additional hardware needed to construct a controller hierarchy with a communication bus in section 5.1. This allows us to perform the desired testing and characterisation operation sequences to the memories with sufficient flexibility to add tests down the line. Next, we will see how all designs combine into a top level design in section 5.2. Finally, we list an initial set of test and characterisation procedures in section 5.3 and conclude this chapter in section 5.4.

5.1. Testing

For testing and characterisation of the memories, sequences of operations need to be applied at relatively high frequencies in order to be fast enough for measuring retention time and achieve a well measurable power consumption. At room temperature, retention times are expected to be in the order of a microsecond, so being able to read and write in the order of $100\,\mathrm{ns}$ is required. Additionally, the expected read and write operation energies derived in chapter 3 are in the range of $100\,\mathrm{fJ}$ to $200\,\mathrm{fJ}$. An operation rate of $10\,\mathrm{MHz}$ then results in a power consumption of $1\,\mathrm{\mu W}$ to $2\,\mathrm{\mu W}$, which can be measured with high accuracy.

The required data rates do not allow us to apply memory operations from outside of the chip. For example, for a write operation, at least five address bits and 32 data bits would be needed, resulting in a data rate of $370\,\mathrm{Mbit\,s^{-1}}$. Especially in a cryogenic measurement setup with several meters of cable between the chip and measurement instruments, these data rates can not be achieved trivially. Additionally, latency measurements are unreliable due to the I/O circuitry in the pad ring which contributes unknown latency.

A simple alternative is to use a Finite State Machine (FSM) that can apply several instruction sequences, generated on-chip and close to the memories. However, this solution lacks flexibility and requires definition of the sequences that are to be tested during the design phase. Since many different tests are possible (see section 5.3), this would lead to a large and complex FSM and leave us unable to test other operation sequences if unexpected results are encountered.

To achieve the desired flexibility, an addressed bus architecture is designed with a global controller that initiates transactions and sends operations on a bus which are interpreted by local controllers. This allows for a single complex global controller design that can be programmed with various instruction sequences and simple local controllers at the memories. This solution gives us the desired memory operation rate and enough flexibility to add test operation sequences even after tape-out.

5.1.1. Bus architecture

The architecture consists of single initiating or global controller, and multiple receiving or local controllers connected to a single shared, addressable bus. As a result, only a single memory can be addressed at the same time. However, this is not an issue as our goal is to characterise each memory individually.

The bus consists of a clock, a reset, an operation code, an address, read and write data, and handshaking signals which are inspired by the handshake in the AXI protocol [73]. The clock is simply the global clock used to synchronise all the digital circuits. The reset signal is a synchronised version of the input reset from the reset I/O pad. In the remainder of this section, we will see the use of the remaining bus signals and what the waveforms look like during a transaction.

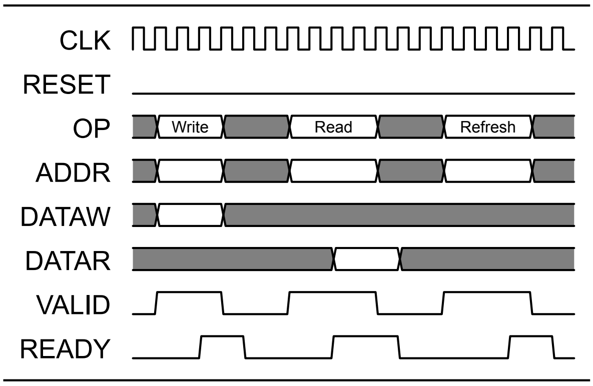
The operation code is a 2 bit signal which selects one of three possible operation. A write operation is encoded as 00, a read operation is encoded as 01, and a refresh operation is encoded as 10. The code 11 also decodes to a read operation to ensure the least intrusive operation is performed in case of an error or glitch.

The address is a 10 bit signal which represents the entire address space seen by the global controller. It is designed such that each row from each memory can be addressed. The four most significant bits are used to select one of at most sixteen local controllers. The five least significant bits are used to address a row in the memory connected to the selected local controller. The remaining bit is used to address the settings register located in the local controller, which is covered in more detail later.

Two 32 bit unidirectional data busses are used for the write and read data. The write data bus is driven by the global controller and received by all local controllers. The read data bus is driven by only the selected memory using tri-state buffers and received by the global controller.

The handshaking signals ensure that transactions of various durations are successfully transmitted and that data is captured before it is removed from the bus. Table 5.1 shows the bus waveforms corresponding with the instructions that can be sent across the bus. A brief description of each transaction is given below, going from left to right in the figure.

Table 5.1: Waveforms on the bus lines showing the handshaking protocol for a write, read, and refresh operations. When a bus is grey, its binary value does not matter and can be changed. When a bus is white, its value must be kept constant by the driver.



Write In case of a write operation, the global controller supplies the operation code, the address, and write data to the bus and asserts the VALID signal to indicate that the instruction is valid. The address and write data are taken from the bus by a local controller and a write operation is executed on its memory if the settings register address bit is 0, or on its settings register if the settings register address bit is 1. Once the write instruction is completed, the READY signal is asserted. The global controller then acknowledges the completion by deasserting the VALID signal, and the transaction is completed by the local controller deasserting the READY signal.

Read In case of a read operation, the global controller supplies the operation code and the address, and asserts the VALID signal to indicate that the instruction is valid. The address is taken from the bus by a local controller and a read operation is executed on its memory, if the settings register address bit is 0, or on its settings register if the settings register address bit is 1. Once the read data is available, the local controller puts it on the bus and asserts the READY signal. The global controller then takes the read data from the bus and deasserts the VALID signal once the data has been copied. The local controller then knows the data is read and can deassert the READY signal to complete the transaction.

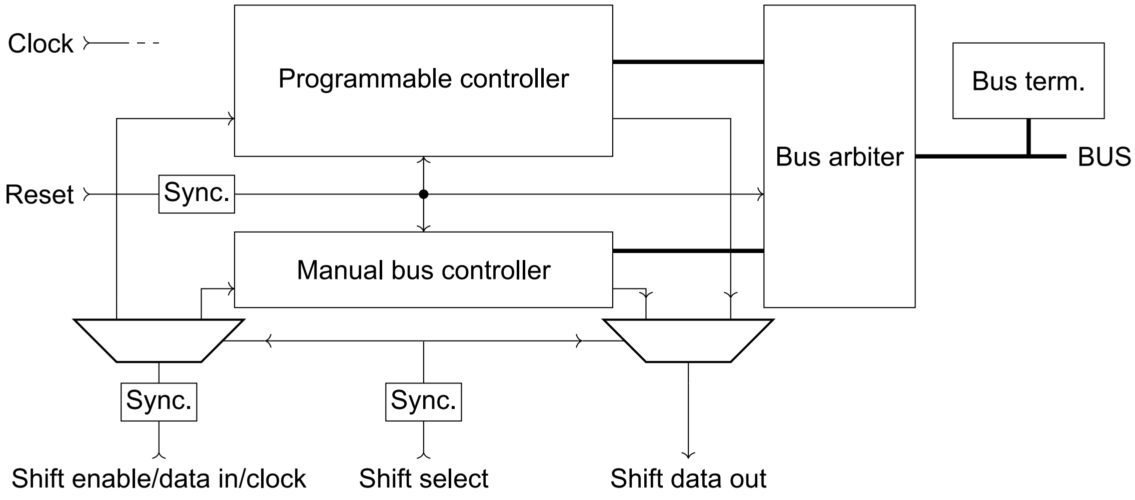


Figure 5.1: Structural architecture of the global controller consisting of a programmable controller and a manual bus controller connected to a bus arbiter. The clock is connected to all blocks, except the bus terminator, and is not shown for clarity.

Refresh In case of a refresh operation, the global controller supplies the operation code and the address, and asserts the `VALID` signal to indicate that the instruction is valid. The address is taken from the bus by a local controller and a read operation is executed, followed by a refresh write operation, if the local controller has a dynamic memory. Once the instructions are completed, the `READY` signal is asserted. If the local controller has a static memory, or the settings register address bit is 1, the instruction is directly acknowledged by assertion of the `READY` signal. The global controller then acknowledges the completion by deasserting the `VALID` signal, and the transaction is completed by the local controller deasserting the `READY` signal.

5.1.2. Global controller

The global controller must be capable of applying memory operations at high speeds, while maintaining a high degree of flexibility. This can be achieved by using a programmable microprocessor with custom instructions to interface with the bus. To provide the ability to manually verify or apply operations, a separate bus controller is added. It can also be used to test microprocessor programs which interface with memories by manually overwriting memory data, simulating a certain memory fault. Since the manual bus controller also initiate a bus transaction, a bus arbiter has to be included to ensure that only one initiator has bus control.

Additionally, a bus terminator is included to ensure that the parts of the bus driven by the receiving controllers are never floating. As mentioned before, the address signal has space for sixteen local controller, but not all addresses are used. If a non-existent address is applied on the address lines, the bus terminator takes control and grounds the read data and `READY` signals.

Finally, since the global controller interfaces with signals generated off-chip, double flip-flop synchronisers are added to synchronise the asynchronous inputs. This leads to the global controller architecture shown in fig. 5.1. In the following sections, we will cover the blocks in more detail.

Microprocessor architecture

The programmable microprocessor follows largely a standard single-cycle processor architecture with a limited instruction set, the block diagram of which is shown in fig. 5.2. Two blocks are added specifically for this application, namely the bus interface and the error counter bank. Additionally, the error counter bank has the power the interrupt execution, which is further explained later. All the memory elements shown in this block diagram are implemented using flip-flops to ensure that these memories work.

Instruction set The instruction memory of the microprocessor supports 32 instructions of 19 bits. The 19 bits encode an instruction following the encoding shown in table 5.2. A description of the different opcodes (OP) is shown in table 5.3. Registers RS1/RS2 indicate the two source registers, while register RD indicates the destination register. IMM and IMMB indicate immediate values for calculations and for program counter (PC) offsets for branch instructions, respectively.

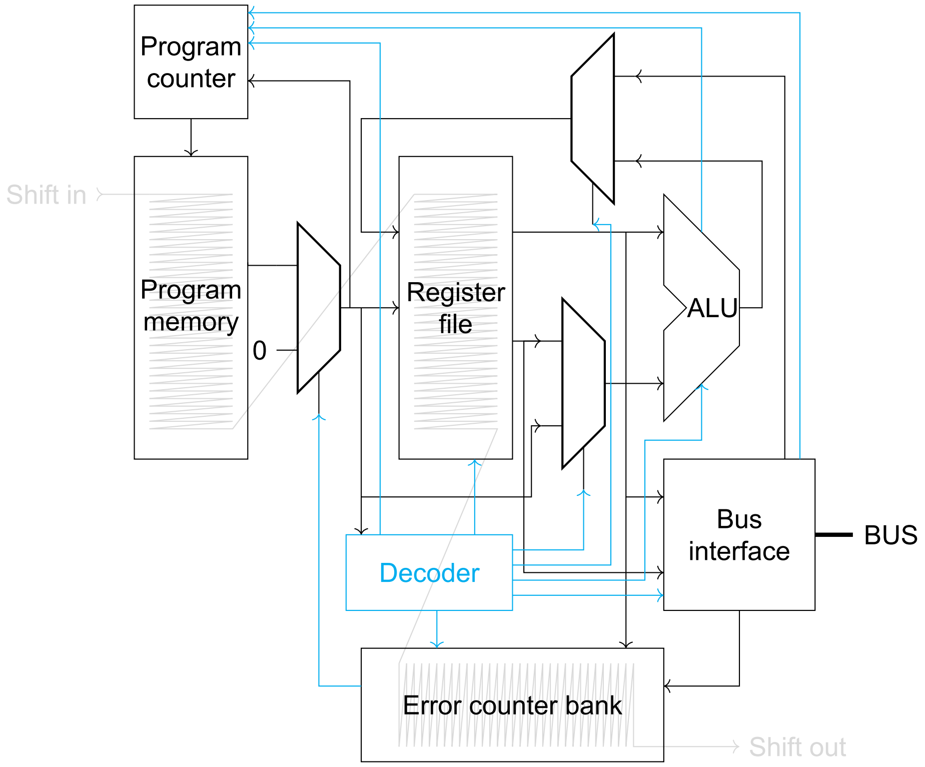


Figure 5.2: Structural schematic of the microprocessor. The data path is shown in black, the control path is shown in cyan, and the shift register is shown in grey. Shift register control signals, reset, and clock are not shown.

Table 5.2: Instructions bit encoding.

Instruction word bits																		
18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP				RD/IMMB						RS2/IMM					RS1			

Although there are 16 different instructions listed in table 5.3, other instructions can be made from these instructions. For example, the no-operation (NOP) instruction can be implemented in multiple ways, by performing an addition with the zero register for all operands, or a 'branch if unequal' operation with the same register for both operands. Also note that the execution of the entire processor can be frozen with an all-zero instruction word. This will cause the PC to stay at its current value until the processor is reset and can be used at the end of a program.

The instructions are decoded by the instruction decoder according to table 5.4. The outputs are used in the program counter, ALU, register file, bus interface, and error counter bank:

- `IS_BRANCH`: used by the PC to determine whether the counter value is incremented by 1 (0) or a new counter value has to be computed with `IMMB`, and optionally selected based on the comparison result of the ALU (1).
- `ALU_OPCODE`: used by the ALU to select its logical function.
- `ALU_OP2_SEL`: used by the ALU to determine whether its second operand comes from the register file (0) or from the `IMM` field of the instruction (1).
- `WRITE_RD`: used by the register file to determine whether the destination register has to be written to (1) or not (0).
- `HALT_ALLOWED`: combined with an output from the bus interface to halt the PC when performing a bus operation.
- `BUS_INT_EN`: indicates whether the bus interface is enabled.

Table 5.3: Description of the instruction for each opcode. The five instructions with opcodes `1000` to `1100` use the bus.

Opcode	Instruction description	Pseudo code equivalent
`0000`	Branch if equal	`if $rs1 = $rs2: PC = PC + immb`
`0001`	Branch if unequal	`if $rs1 != $rs2: PC = PC + immb`
`0010`	Branch if less than	`if $rs1 < $rs2: PC = PC + immb`
`0011`	Branch if less than or equal	`if $rs1 <= $rs2: PC = PC + immb`
`0100`	2s complement register add	`$rd = $rs1 + $rs2`
`0101`	Bitwise register XOR	`$rd = $rs1 XOR $rs2`
`0110`	Register shift left	`$rd = $rs1 << $rs2[4:0]`
`0111`	Register shift right	`$rd = $rs1 >> $rs2[4:0]`
`1000`	Write over bus	`write(data = $rs1, address = $rs2[9:0])`
`1001`	Refresh over bus	`refresh(address = $rs2[9:0])`
`1010`	Read and store	`$rd = read(address = $rs2[9:0])`
`1011`	Read, compare, accumulate	`errors += (($rd = read(address = $rs2[9:0])) != $rs1)`
`1100`	Read, compare, halt	`if ($rd = read(address = $rs2[9:0])) != $rs1: halt`
`1101`	Sign extended imm. add	`$rd = $rs1 + imm`
`1110`	Immediate shift left	`$rd = $rs1 << imm`
`1111`	Immediate shift right	`$rd = $rs2 >> imm`

- `BUS_INT_OP`: used by the bus interface to select the bus operation.
- `ERR_CNT_EN`: indicates whether the error counter bank is enabled.
- `ERR_CNT_OP`: used by the error counter bank to select whether the errors need to be accumulated, or execution should be halted once an error is encountered.
- `READ_OP`: indicates whether the bus operation is a read operation to ensure that the register file waits until the read operation is completed before capturing the read data.

Bus interface The bus instructions cause the bus interface to send a message on the bus and temporarily halt the processor until the transaction is completed. When a bus instruction is encountered, the decoder triggers the bus interface FSM, shown in fig. 5.3, to start the bus transaction sequence. The `HALT` output is combined with the instruction decoder output `HALT_ALLOWED` using an AND-gate. This combination will halt the PC from the moment the bus instruction enters the decoder until the bus interface FSM deasserts its `HALT` output. The `READ_VALID` output is used by the register file to store the data from a read instruction after it has been received.

The bus interface contains registers at the bus side to ensure that the bus outputs are stable and the bus inputs have enough time to settle. The structural architecture of the bus interface is shown in fig. 5.4. The output registers are updated when a bus instruction is applied (`SEND=1`) and the `READY` signal is not yet set. The input registers are always updated and contain the latest values from the bus.

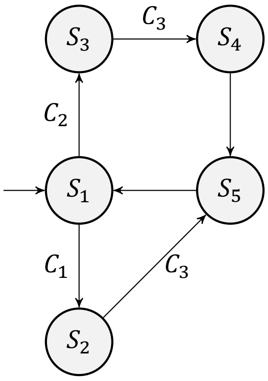
Error counters The error counters are responsible for storing the comparison results of the read and compare instructions. In both cases, the read data is compared bitwise with the data in a specified register. The error counter bank contains 32 $10\,\mathrm{bit}$ registers which can be used to accumulate the amount of differences encountered during multiple read operations for every bit position with read, compare, and accumulate instructions.

The error counters can also be used to halt execution once a single error has been encountered. A `STOP` register that can only be reset using the `RESET` signal, is set once an error is encountered during the read, compare, and halt instruction. The output of the `STOP` register can overwrite the program memory output with an all-zero instruction word, as shown in fig. 5.2, which will cause the entire processor to freeze until it is reset again.

Programming and readout Programming the processor and reading out the results is done through a shift register that passes through the program memory, the register bank, and the error counters, as indicated in fig. 5.2. This results in a total shift register length of $32\times19+31\times32+32\times10 = 1920\,\mathrm{bit}$.

Table 5.4: Instruction decoder outputs for each opcode.

Opcode	IS_BRANCH	ALU_OPCODE	ALU_OP2_SEL	WRITE_RD	HALT_ALLOWED	BUS_INT_EN	BUS_INT_OP	ERR_CNT_EN	ERR_CNT_OP	READ_OP
0000	1	COMP_EQ	0	0	0	0	WRITE	0	ACC	0
0001	1	COMP_NEQ	0	0	0	0	WRITE	0	ACC	0
0010	1	COMP_LT	0	0	0	0	WRITE	0	ACC	0
0011	1	COMP_LTEQ	0	0	0	0	WRITE	0	ACC	0
0100	0	ADD	0	1	0	0	WRITE	0	ACC	0
0101	0	XOR	0	1	0	0	WRITE	0	ACC	0
0110	0	SHL	0	1	0	0	WRITE	0	ACC	0
0111	0	SHR	0	1	0	0	WRITE	0	ACC	0
1000	0	COMP_EQ	0	0	1	1	WRITE	0	ACC	0
1001	0	COMP_EQ	0	0	1	1	REFRESH	0	ACC	0
1010	0	COMP_EQ	0	1	1	1	READ	0	ACC	1
1011	0	COMP_EQ	0	1	1	1	READ	1	ACC	1
1100	0	COMP_EQ	0	1	1	1	READ	1	HALT	1
1101	0	ADD	1	1	0	0	WRITE	0	ACC	0
1110	0	SHL	1	1	0	0	WRITE	0	ACC	0
1111	0	SHR	1	1	0	0	WRITE	0	ACC	0



		Outputs		
State	Name	HALT	READ_VALID	VALID
S_1	Reset	1	0	0
S_2	Wait for ready (write/refresh)	1	0	1
S_3	Wait for ready (read)	1	0	1
S_4	Read data	1	1	1
S_5	Continue processor	0	0	0

Condition	Description
C_1	SEND = 1 & READY = 0 & OP != read
C_2	SEND = 1 & READY = 0 & OP = read
C_3	READY = 1

Figure 5.3: State diagram of the bus interface FSM. If a none of the departing conditions are met, the state is held.

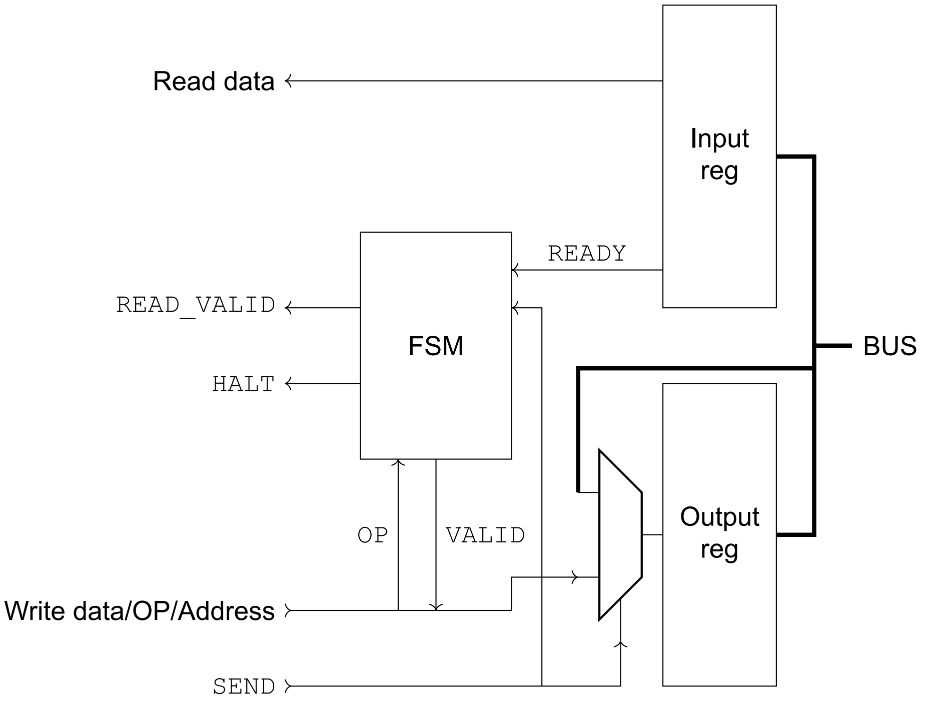


Figure 5.4: Structural schematic of the bus interface.

During programming, this allows us to initialise the registers and error counters to desired values without having to spend instructions to do so. During readout, the program and its results are automatically combined, which allows for easy documentation of results and the program that produced them.

The shift register design uses three control signals, a data input signal, and a data output signal. A selection control signal is used to select either the shift register from the programmable controller or from the manual bus controller. A shift enable signal puts the shift register in shifting mode, connecting all the registers in a chain. A shift clock signal drives an edge detector that will cause the shift register to shift a single position on a rising edge. This allows the shift clock to be any speed, so we can ensure functionality even with long interconnect paths such as in the cryogenic measurement setup.

The processor can be read out when the program has finished, or when the program has halted the processor. As mentioned before, the processor can be halted using an all-zero instruction word. Additional logic is added to compare the current instruction (after the multiplexer with the all-zero instruction word) with the all-zero instruction word. When the shift register is not enabled, the shift out signal is overwritten by the result of this comparison. This allows us to see if the program has terminated and the results are ready to be shifted out.

Manual bus controller

The manual bus controller is programmed and read out using a second shift register and controlled by a simple FSM. The architecture is shown in fig. 5.5 and the FSM state diagram is identical to the one shown in the bus interface in fig. 5.3, except the `HALT` output is inverted to `VALID_CLEAR`, the `VALID` output is removed, and the `SEND` input is driven by the `VALID` register output. The bus operation code, address, data, and valid registers are set using the shift register, resulting in a total shift register length of 45 bit. The valid register is at the end of the shift register and therefore allows us to see whether the operation is completed yet, similar to the programmable controller. The FSM determines when the valid field is deasserted, depending on the bus operation code, and allows the data field to be overwritten in case of a read operation.

Bus arbiter

The bus arbiter ensures that the processor and manual bus controller do not drive the bus at the same time. This is implemented using a two-state FSM, which is shown in fig. 5.6. By default, the processor is connected to the bus (`INPUT_SELECT = 0`). Only when the manual bus controller want to make a

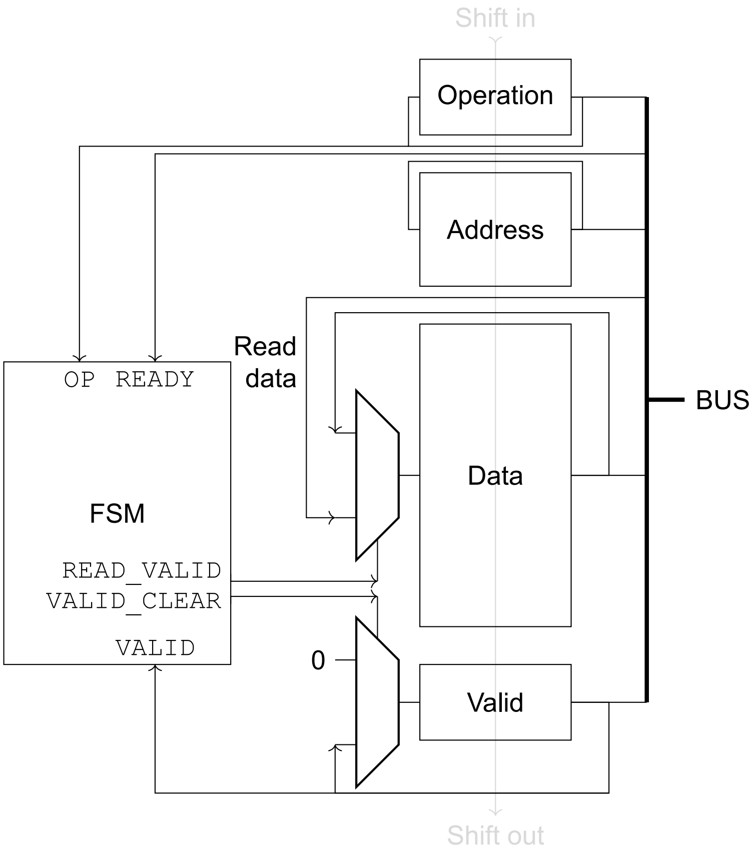
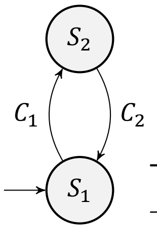


Figure 5.5: Structural schematic of the manual bus controller. The grey line indicates the shift register.



State	Name	Outputs `INPUT_SELECT`
S_1	Programmable controller	0
S_2	Manual controller	1

Condition	Description
C_1	`VALID_0 = 0 & READY = 0 & VALID_1 = 1`
C_2	`VALID_1 = 0 & READY = 0`

Figure 5.6: State diagram of the bus arbiter FSM.

bus transaction (`VALID_1 = 1`) and the bus is free (`VALID_0 = 0 & READY = 0`), is the bus handed to the manual bus controller (`INPUT_SELECT = 1`). Once its transaction is finished (`VALID_1 = 0 & READY = 0`), the bus is immediately handed back to the processor. This minimises the additional bus delay experienced by the processor since bus negotiation is almost never needed.

Implementation

The global controller is synthesised using Cadence Genus and implemented using Cadence Innovus. Additional hold margins of $50\,\mathrm{ps}$ are included to ensure that the logic will still work at cryogenic temperatures. The logic is expected to become faster at cryogenic temperatures, which is especially dangerous for hold requirements as they can not be fixed by changing the clock frequency, contrary to setup violations. Additionally, the density of well taps has been increased to a well tap spacing of $8\,\mathrm{\mu m}$ to reduce the latch-up risks associated with the increased substrate resistance. This spacing has been used in previous cryogenic logic designs in which no latch-up issues have been encountered.

To obtain a memory operation frequency of about $10\,\mathrm{MHz}$, the system has to run at $100\,\mathrm{MHz}$. Including loop and bus transaction overhead, about ten clock cycles are needed for a memory operation to complete. The timing constraints at a clock frequency of $150\,\mathrm{MHz}$ are satisfied for both synthesis and implementation, which means that the clock frequency could even be increased to push the memory operation frequency higher during testing.

5.1.3. Local controller

Each memory will be accompanied by a local controller which monitors the bus and executes memory operations that are targeted at its memory. The local controllers also store settings for their memories, such as delay chain settings or the sense amplifier test mode. Additionally, a launch-capture register pair is included to measure the read latency of each memory.

Architecture and FSM

The local controller consists of an FSM which is responsible for interpreting the bus command and controlling the data flow between the bus, memory, and several registers. The architecture of the general local controller is shown in fig. 5.7. Depending on the application of the local controller, not all registers or functionalities are included. The bus interfaces on the left, the memory interfaces on the right. The address comparator compares the upper four address bits with its local address to determine if it is selected. If it is selected, the bus output tri-state buffers are enabled. The different sections of the settings register are decoded using static decoders into the format needed by, for example, the delay chains.

The FSM diagram is shown in fig. 5.8, where the application dependent parts are indicated. For example, the local controller connected to the static memory does not support a refresh instruction and instead skips the blue states (S_6 and S_7) and immediately acknowledges the instruction by going to the 'write done' state (S_4). Similarly, the local controller accompanying the static cell SNM array does not support any memory operations and only has a settings registers to enable selected transmission gates. In that case, only the grey states are implemented.

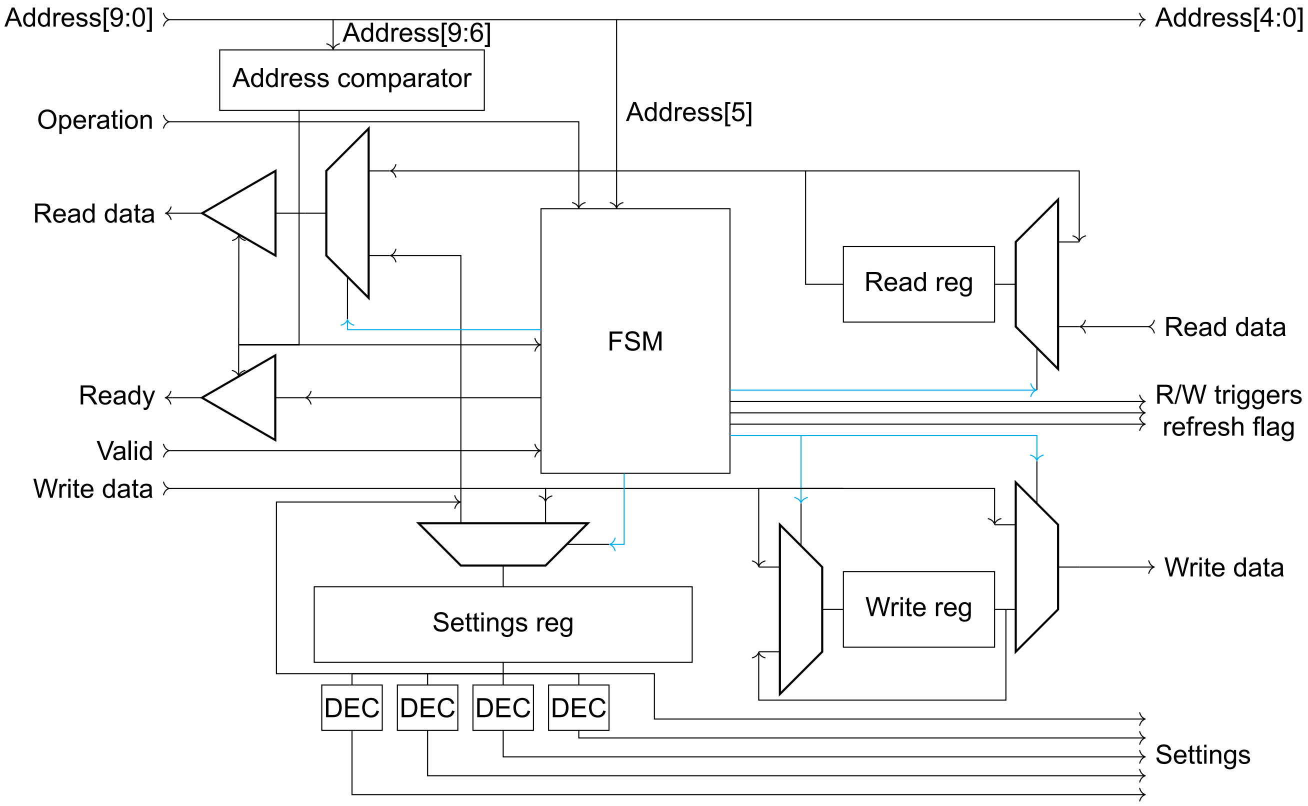


Figure 5.7: Structural schematic of a local controller.

Measuring latency

To measure the total latency of a read operation, a launch-capture-synchronise register setup is used as shown in fig. 5.9. At the rising clock edge, the inputs to the memory are updated. After a delay set by a delay chain (ΔT), the outputs of the memory are clocked. If the latency of the memory is lower than the $\Delta t - t_{\text{clk-to-q}} - t_{\text{setup}}$, the outputs are correctly captured, otherwise the previous value is still captured. By performing a read operation with a different expected output than the previous read, errors can be detected if ΔT is too small.

To ensure that even results from the maximum read duration can be captured, a delay chain with 256 steps is used, resulting in a maximum delay of approximately $5.12\,\mathrm{ns}$. This way, the entire latency measuring system can effectively be disabled by setting it to the maximum delay possible.

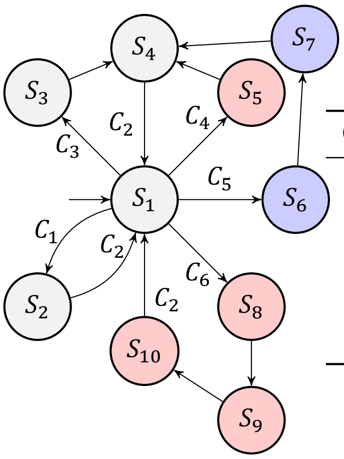
To characterise the delay chain, an exact copy of the 256-step delay chain in a ring oscillator (RO) configuration is included in the design. This is required to measure the delay chain's setting-to-delay mapping at room temperature and cryogenic temperatures and correct for changes in stage delay at different temperatures. This ring oscillator is attached to a local controller with only a settings register, similar to the SNM array. This allows us to vary the delay of the chain by sending write commands on the bus, from which the setting-to-delay mapping can be approximated. See section 5.3 for more details on the test procedure.

Settings register

The local controllers contain a settings register which stores the memory and testing settings. The mapping of these settings varies between the local controller types, as shown in table 5.5. This table shows what each bit from every local controller is used for. The maximum size for the settings register is $32\,\mathrm{bit}$, but only the required length is actually implemented to save area.

Implementation

Like the global controller, the local controllers are also synthesised using Cadence Genus and implemented using Cadence Innovus with an additional hold margin of $50\,\mathrm{ps}$ and high well tap density. Timing constraints are met with a target clock frequency of $500\,\mathrm{MHz}$. This is larger than the intended



Condition	Description
C_1	VALID & SELECTED & ADDR[5] & OP != write
C_2	!VALID
C_3	VALID & SELECTED & ADDR[5] & OP = write
C_4	VALID & SELECTED & !ADDR[5] & OP = write
C_5	VALID & SELECTED & !ADDR[5] & OP = refresh
C_6	VALID & SELECTED & !ADDR[5] & OP = read

State	Name	Outputs							
		SET_W	SET_R	TRIG_W	LOAD_W	REF_FLAG	TRIG_R	LOAD_R	READY
S_1	Reset	0	0	0	0	0	0	0	0
S_2	Read settings	0	1	0	0	0	0	0	1
S_3	Write settings	1	0	0	0	0	0	0	0
S_4	Write done	0	0	0	0	0	0	0	1
S_5	Write	0	0	1	1	0	0	0	0
S_6	Refresh read	0	0	0	0	0	1	0	0
S_7	Refresh write	0	0	1	0	1	0	0	0
S_8	Read	0	0	0	0	0	1	0	0
S_9	Read memory	0	0	0	0	0	0	1	0
S_{10}	Read done	0	0	0	0	0	0	0	1

Figure 5.8: General state diagram of the local controller FSM. The local controllers for the dynamic memories use the entire diagram. The blue states are removed in local controllers for the static memories since they can not refresh. The red and blue states are removed in the local controller for the SNM array and delay chain ring oscillator (see section 5.1.3) since they only need a settings register.

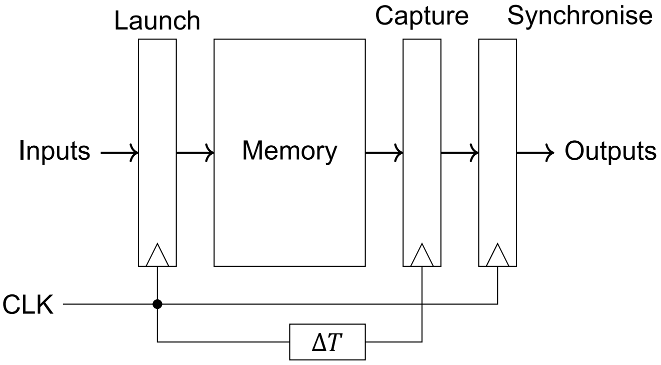


Figure 5.9: Launch-capture-synchronise register latency measuring setup. If the latency of the memory is less than $\Delta T - t_{\text{CQ}} - t_{\text{setup}}$, the latency from input to output is one clock cycle. If the latency is larger, the input to output latency is two clock cycles.

Table 5.5: Allocation of settings register of different local controllers.

Bit	Dynamic memory	Static memory	SNM array	RO
0	256-step capture register delay chain setting	256-step capture register delay chain setting	32 row select	256-step RO delay chain setting
1				
2				
3				
4				
5			16 column select	
6				
7				
8	192-step read delay chain setting	192-step read delay chain setting		Enable RO
9			Enable bitline	-
10			Bitline data	-
11			Test source follower	-
12			-	-
13			-	-
14			-	-
15			-	-
16	16-step read delay chain setting	16-step read delay chain setting	-	-
17			-	-
18			-	-
19			-	-
20	64-step write delay chain setting	64-step write delay chain setting	-	-
21			-	-
22			-	-
23			-	-
24			-	-
25			-	-
26	Sense amplifier test mode	Sense amplifier test mode	-	-
27	Enable constant write bitline background	16-step write delay chain setting	-	-
28	Constant write bitline background data		-	-
29	-		-	-
30	-		-	-
31	-	-	-	-

clock frequency of $100\,\mathrm{MHz}$, allowing for a higher clock frequency to increase the memory operation frequency beyond $10\,\mathrm{MHz}$.

The layout of the launch and capture registers and delay chain is done manually to ensure that all path lengths and loads are equal. This ensures that the delay for the different bits is roughly equal and that the layout of the registers is efficient within the bitline pitch. The delay chain design is the same as the delay chains used for the memory timing signal generation to ensure matching between the different delay chains such that similar settings lead to similar delays everywhere. The delay between the capture and synchronise register is not critical, so the synchronise register is synthesised and implemented together with the remainder of the local controller.

The layout of the 2T NW-PR cell memory design with the latency measurement hardware and a local controller is shown in fig. 5.10 (standard cell layouts are obfuscated). The registers and delay chain are indicated using the coloured boxes. For all other memories, the positions of the registers and delay chain are the same. The layout of the registers is stretched for the other memories, such that it matches the bitline pitches.

5.2. Top level

The controllers and memories are combined to form the entire system as shown in fig. 5.11. The addresses of the local controllers are shown and run from 0 to 9. Starting from address 10, the bus terminator of the global controller takes care of the bus to prevent a floating bus.

5.2.1. Clock and data distribution

Since the entire system is synchronous, a single clock has to be delivered to all the controllers and data must traverse the bus fast enough. This is achieved by using a custom clock tree, the schematic of which is shown in fig. 5.12. The interconnect between the clock buffers consists of coaxial lines with double width and double spacing with a grounded shield. This results in a high clock line capacitance, but also keeps it constant and shields the other circuits from clock interference due to coupling capacitances. The global controller driving inverter is small since the global controller has a clock tree of its own. Since local controllers eight and nine don't use capture and launch registers, the total amount of clock interconnect is much lower, allowing for a weaker buffer.

To prevent clocking issues and speed up the bus transaction, the clock of the global controller is inverted with respect to the local controllers. This makes the system more resilient against the clock skew between the global controller and the local controllers, which can be seen in the simulation results in fig. 5.13. Since the skew is significant, about $200\,\mathrm{ps}$, fast messages on the bus may catch up with the clock and cause setup or hold issues at the receiving end. By inverting the clock between global and local controllers, the bus gets approximately $5\,\mathrm{ns}$ to settle and is unaffected by skew between the controllers. The only requirement for this to work is that the bus delay is less than roughly $4\,\mathrm{ns}$. However, if this was not the case, then inverting the clock would not be needed since the skew is then much less than the bus delay. Finally, inverting the clocks between the controllers also leads to shorter memory instruction durations, as the handshake overhead now only takes one clock cycle instead of two.

To distribute the data, a $79\,\mathrm{bit}$ bus is used without special shielding. Figure 5.14 shows the delay of the write lines of the bus with a maximum strength driver standard cell, where the fifteenth bit pulses high while all other lines pulse low. Depending on how the neighbours of a line are switching, the delay varies between $70\,\mathrm{ps}$ (all in the same direction) and $500\,\mathrm{ps}$ (all in the opposite direction). This is fast enough for the inverted clock scheme mentioned before as it is less than $4\,\mathrm{ns}$ and so fast that it could indeed lead to data being faster than the clock ($200\,\mathrm{ps}$).

The controller system with inverted clocks at $100\,\mathrm{MHz}$ is simulated in Siemens QuestaSim using maximum post-route SDF timing files with an additional $1\,\mathrm{ns}$ bus delay on top of the bus delays specified in the SDF files. This does not result in any setup violations. Additionally, the same simulation using minimum post-route SDF timing files and without any additional bus delay on top of the SDF bus delay does not result in any hold violations. This indicates that the entire system is fast enough at $100\,\mathrm{MHz}$, and enough delay has been inserted by the implementation tool to slow down the fastest paths.

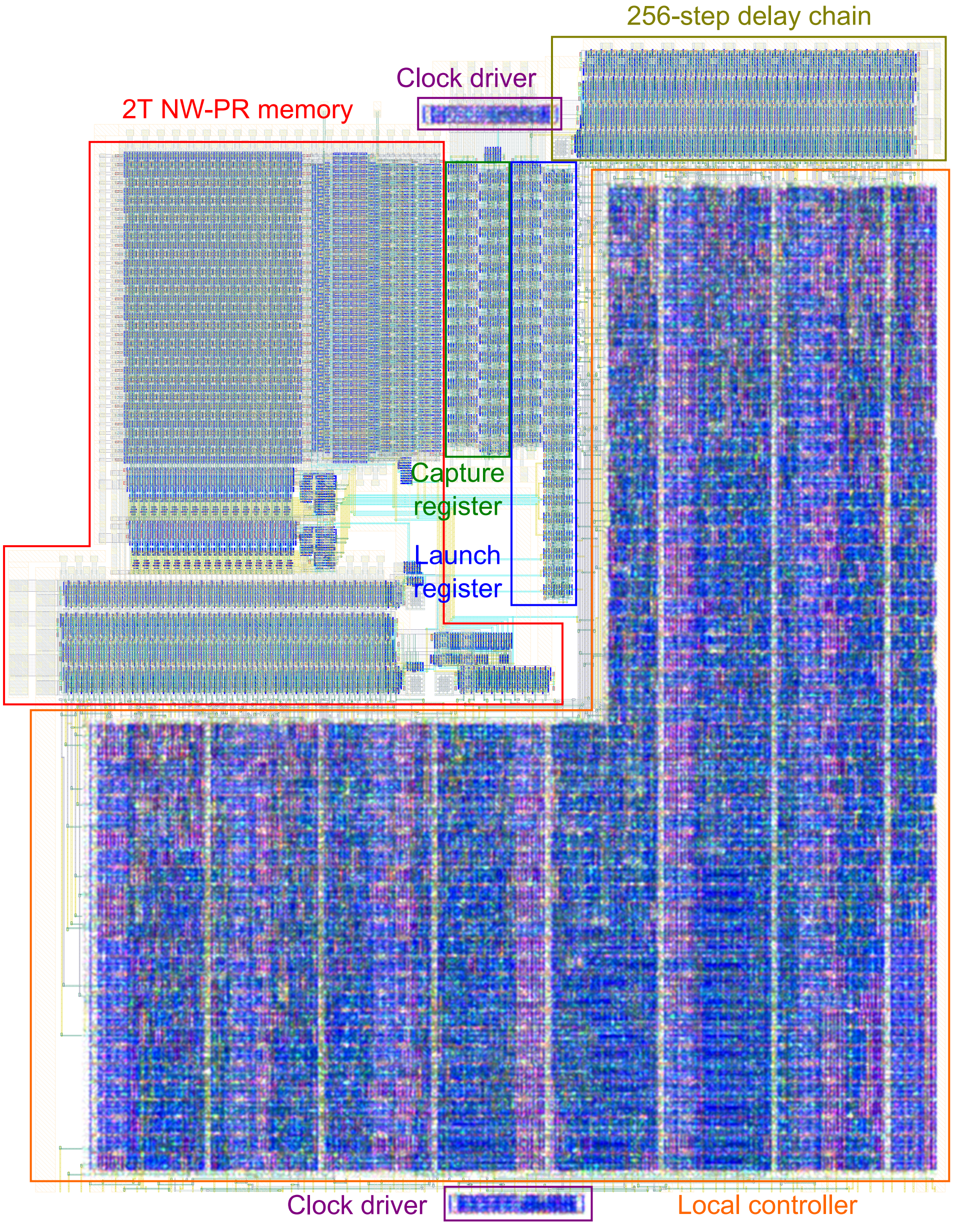


Figure 5.10: Layout of 2T NW-PR memory with launch and capture registers, delay chain, synthesised and implemented local controller (obfuscated), and clock drivers (obfuscated). The bus connects at the bottom of the layout to the local controller.

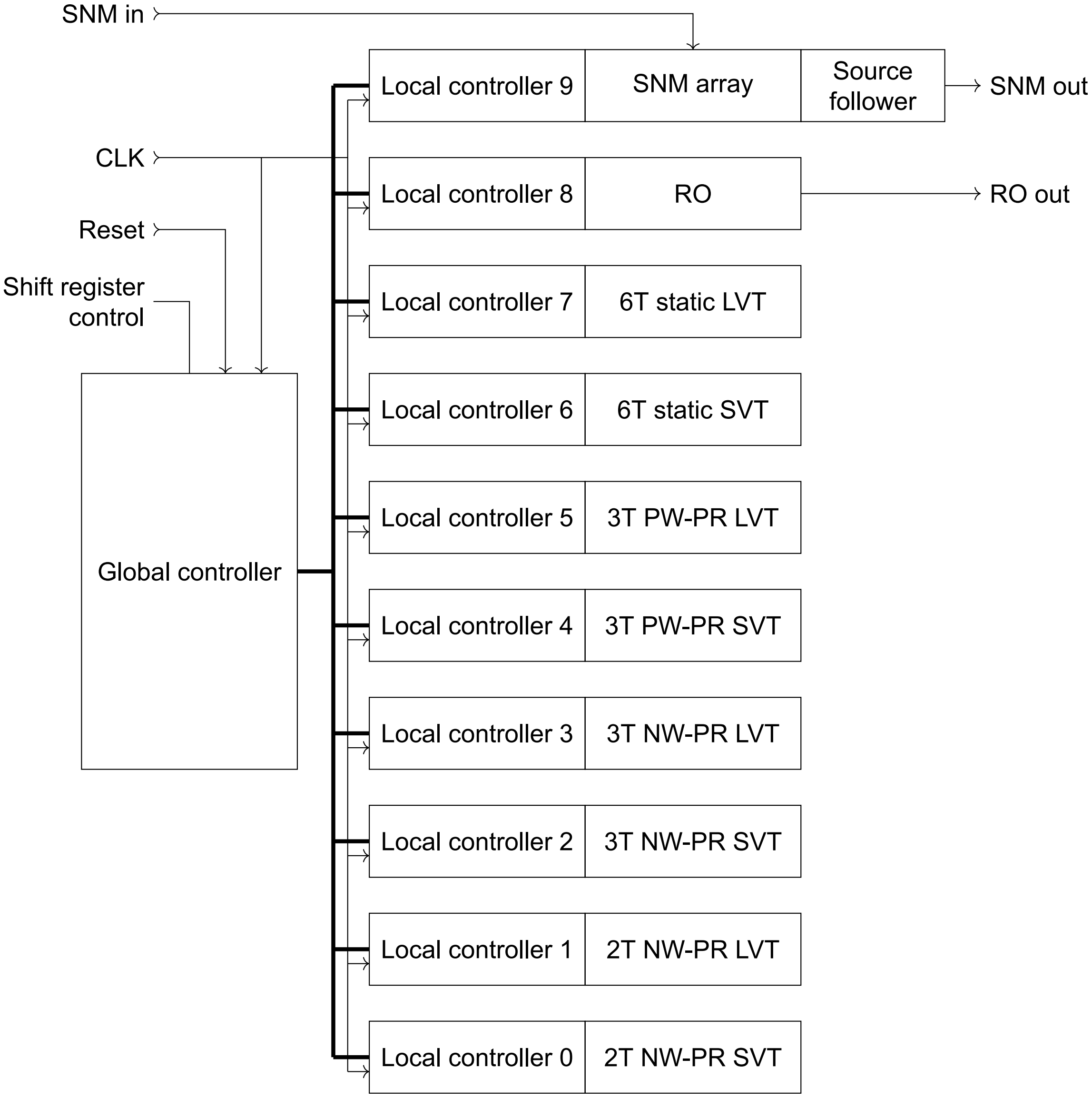


Figure 5.11: Structural schematic of top level memory system. One global controller and ten local controllers are connected through a bus and clock network.

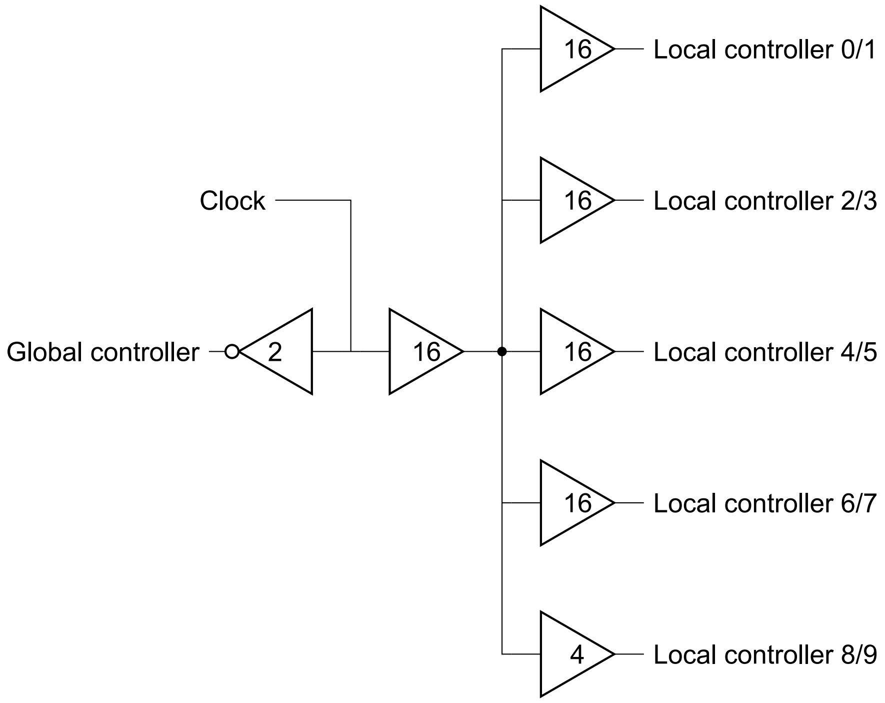


Figure 5.12: Schematic of clock tree for top level memory system. The numbers in the cells indicate the cell output drive strength.

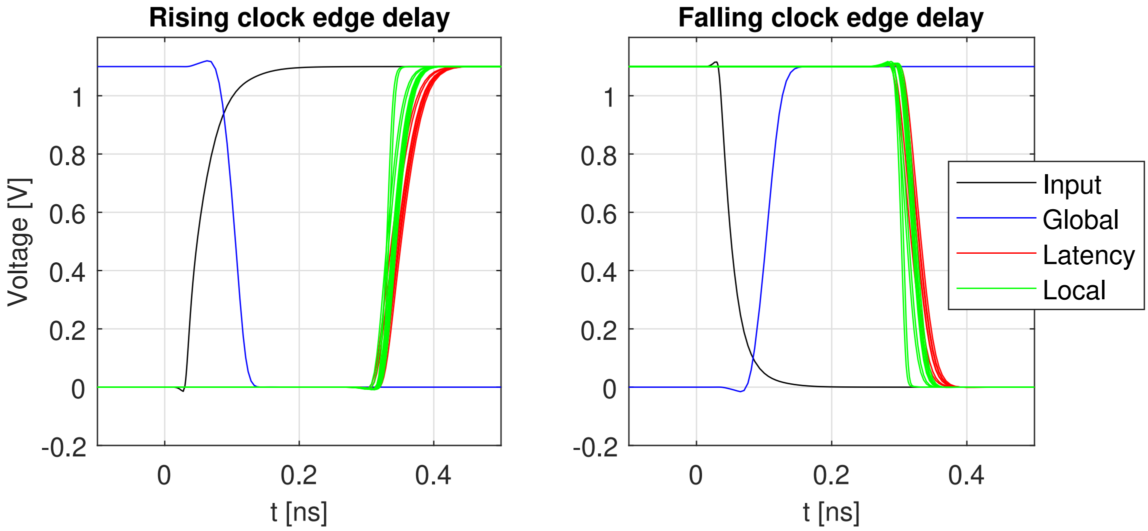


Figure 5.13: Simulation of custom clock tree including layout parasitics and realistic loads. The black line shows the clock at the input of the tree, blue at the input of the global controller, red at the inputs of the capture registers for latency measurement, and green at the inputs of the local controllers.

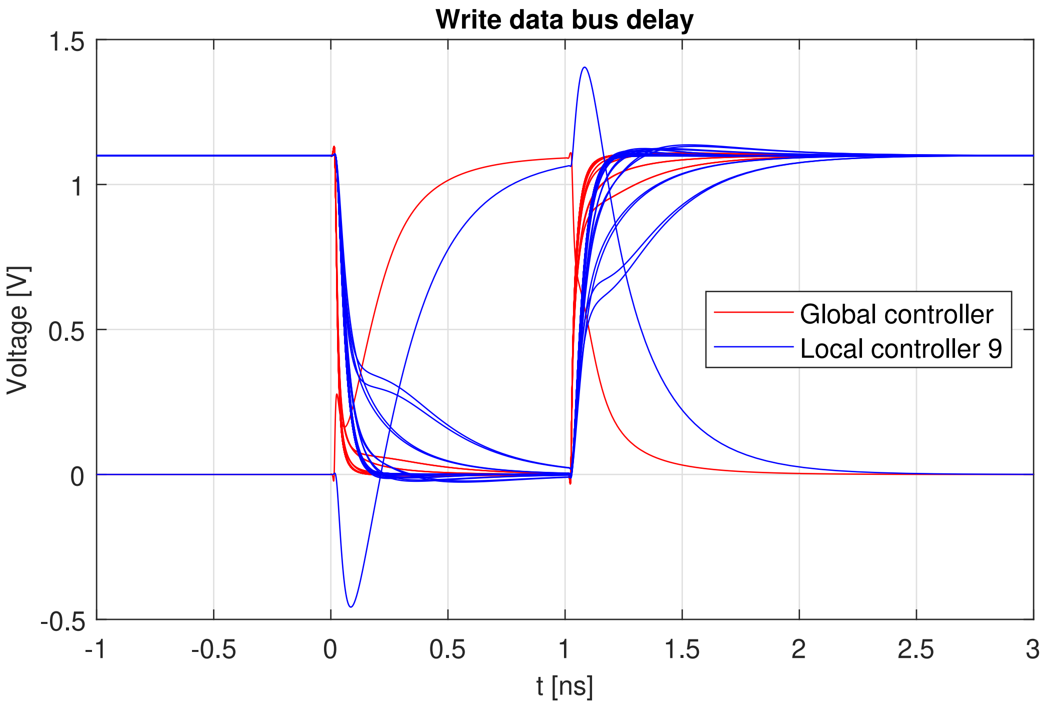


Figure 5.14: Simulation of write bus, including layout parasitics, with realistic loads and drivers. The red lines show the bus voltages at the driving global controller, the blue lines show the bus voltages at the receiving local controller furthest from the global controller (local controller 9).

5.2.2. Floating gate device

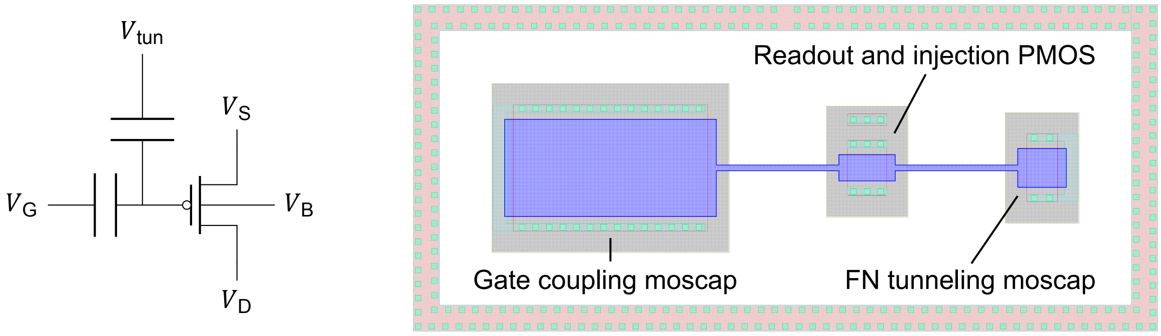
Since the controllers and memories do not completely fill the chip, a floating gate device is also included for separate characterisation. A floating gate device acts similar to a flash cell in that it stores data as charge on a completely floating node through tunnelling and hot carrier injection [74], [75]. Since the gate is fully surrounded by gate and field dielectric, charge can leak away only very slowly, resulting in long term and even non-volatile storage possibilities. Additionally, it can be used to store analog values for various applications such as amplifier offset compensation [76], DACs [77], or programmable voltage and current references [78], [79]. Applications in CMOS quantum control circuits have also been proposed in literature [80].

The schematic of the floating gate device is shown in fig. 5.15a. The corresponding layout is shown in fig. 5.15b, where the transistor and capacitors measure $6.04\,\mu\mathrm{m} \times 1.74\,\mu\mathrm{m}$. Both are heavily inspired by the schematic and layout shown in [74]. The large capacitor on the left side is the gate coupling capacitor. It is large to ensure that the floating gate voltage approximately follows the applied voltage V_G and is used to change the gate voltage of the PMOS transistor at the middle. The small capacitor on the right side is the tunnelling capacitor. By applying a large voltage across it, in the order of $5\,\mathrm{V}$ to $6\,\mathrm{V}$, Fowler-Nordheim tunnelling can be induced, which moves charge onto the floating gate and increases its voltage. The PMOS transistor in the middle is used for readout, because its current is affected by the amount of charge on the floating node. Additionally, by applying a large voltage between its drain and source, again in the order of $5\,\mathrm{V}$ to $6\,\mathrm{V}$, high energy holes are created which can generate high energy electrons on impact that can tunnel through the gate oxide and onto the floating gate. This can be used to lower the gate potential.

The five terminals of the floating gate device are directly connected to analog pads for testing and characterisation. This keeps the design simple and gives full control over the applied terminal voltages. A brief description of the test plan for this device can be found in section 5.3.

5.2.3. Pad ring

The pad ring facilitates the interface between the on-chip and off-chip circuits, and protects the on-chip circuits from ESD events. Figure 5.16 shows a schematic view of the pad ring where three pad ring



(a) Schematic

(b) Layout

Figure 5.15: Schematic and layout of the floating gate device.

sections are clearly visible, separated by power-cuts. The digital part of the I/O ring contains digital pads for the clock, reset, shift register control, and RO output, and supply pads for the digital logic and memories. The core voltage analog part of the I/O ring contains several analog inputs such as the reference voltage inputs and the input for the SNM array that do not require voltages higher than $1.1\,\mathrm{V}$. The I/O voltage analog part of the I/O ring contains pads for the source follower for the SNM array output and the floating gate control pads. Additionally, each section contains supply and ground pads for the ESD protection.

In order to measure the power consumption of the memory cores separately from the timing generation circuits and the remaining digital circuits, multiple supply pads are used. All dynamic memories are connected to the same supply, while the static memories have separate pads. This allows for a static leakage measurement on each of the static memories separately. The timing generation circuits are connected to a separate supply such that their power consumption can also be measured separately, and the timing could be slightly adjusted by varying the supply voltage.

To prevent oscillations due to LC tanks consisting of the bondwire inductance and on-chip decoupling capacitance, degeneration resistors are added to dampen oscillations. These are placed on seven supplies and inputs from which current pulses are drawn. These resistors are implemented using unsalicided polysilicon resistors and have a resistance of $30\,\Omega$ to $40\,\Omega$.

5.2.4. Complete layout

Figure 5.17 shows the layout of the completed chip, which measures $1085\,\mu\mathrm{m}\times1085\,\mu\mathrm{m}$. The coloured boxes indicate the different components. A large part of the chip is filled with decoupling capacitance to ensure that voltages are constant, despite the spiking current consumption of the various circuits. The various memory types are shown and the local controllers are numbered according to their local addresses.

5.2.5. Verification

The test architecture has been verified at various stages during the design. The controller hierarchy has been simulated in combination with a behavioural memory model in VHDL, including the post-routing SDF timing files. Additionally, the system has been compiled onto an Digilent Nexys 4 FPGA development board with a Xilinx ARTIX-7 FPGA and shown to work while using the FPGA's BRAM blocks as memories connected to the local controllers. Finally, the entire top level schematic, up to and including the pad ring, has been simulated using Cadence Spectre which showed that write, read, and refresh operations on individual memories work as expected.

5.3. Test plan

In this section, an overview of some of the measurements and experiments that can be done with this chip is presented. Due to the flexibility in the programmable microprocessor, the presented set of tests can always be expanded. In the following sections, an initial set of measurements is presented to extract various characteristics and performance metrics from the memories and individual devices.

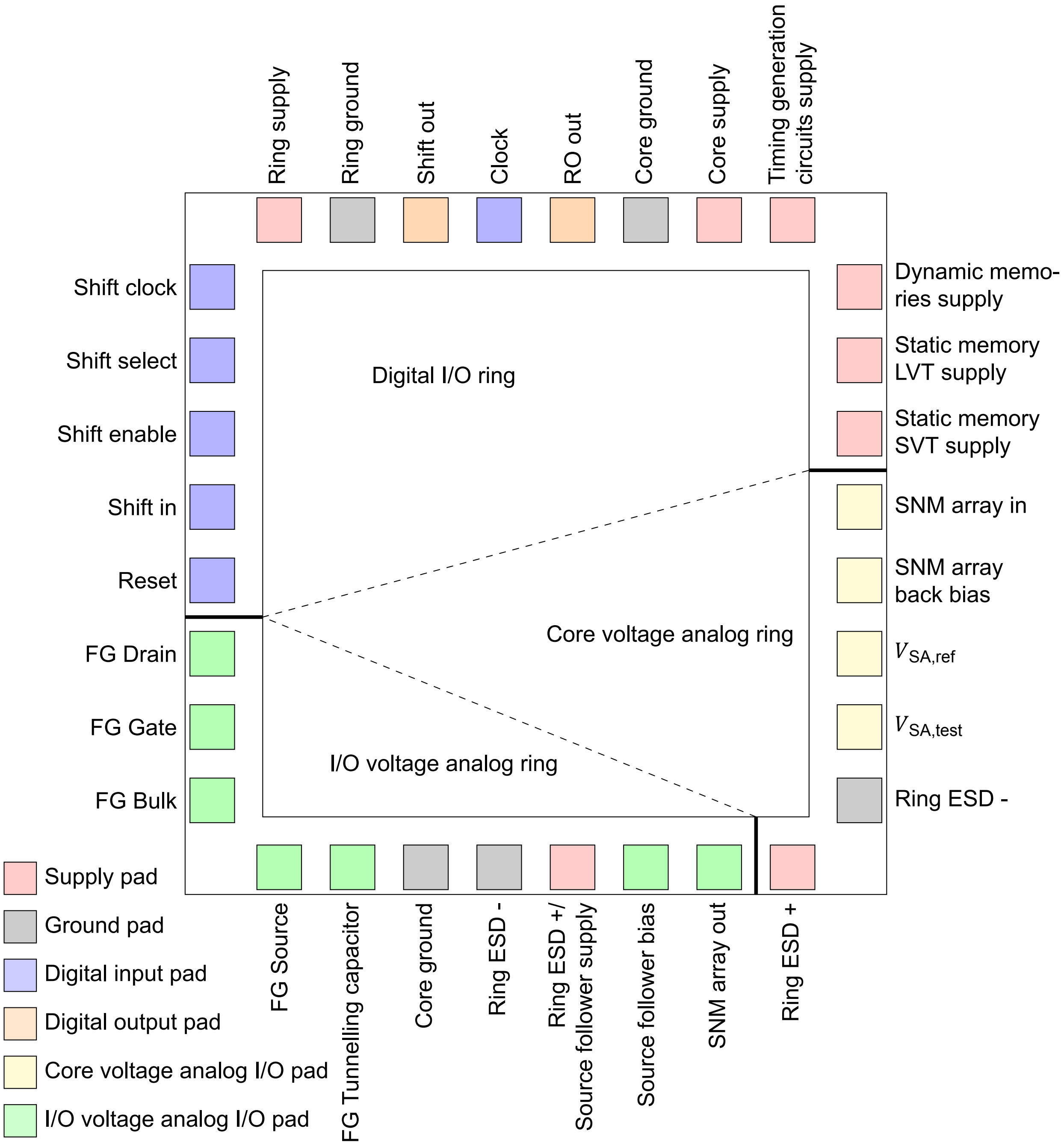


Figure 5.16: Schematic of the pad ring with colour-coded pads.

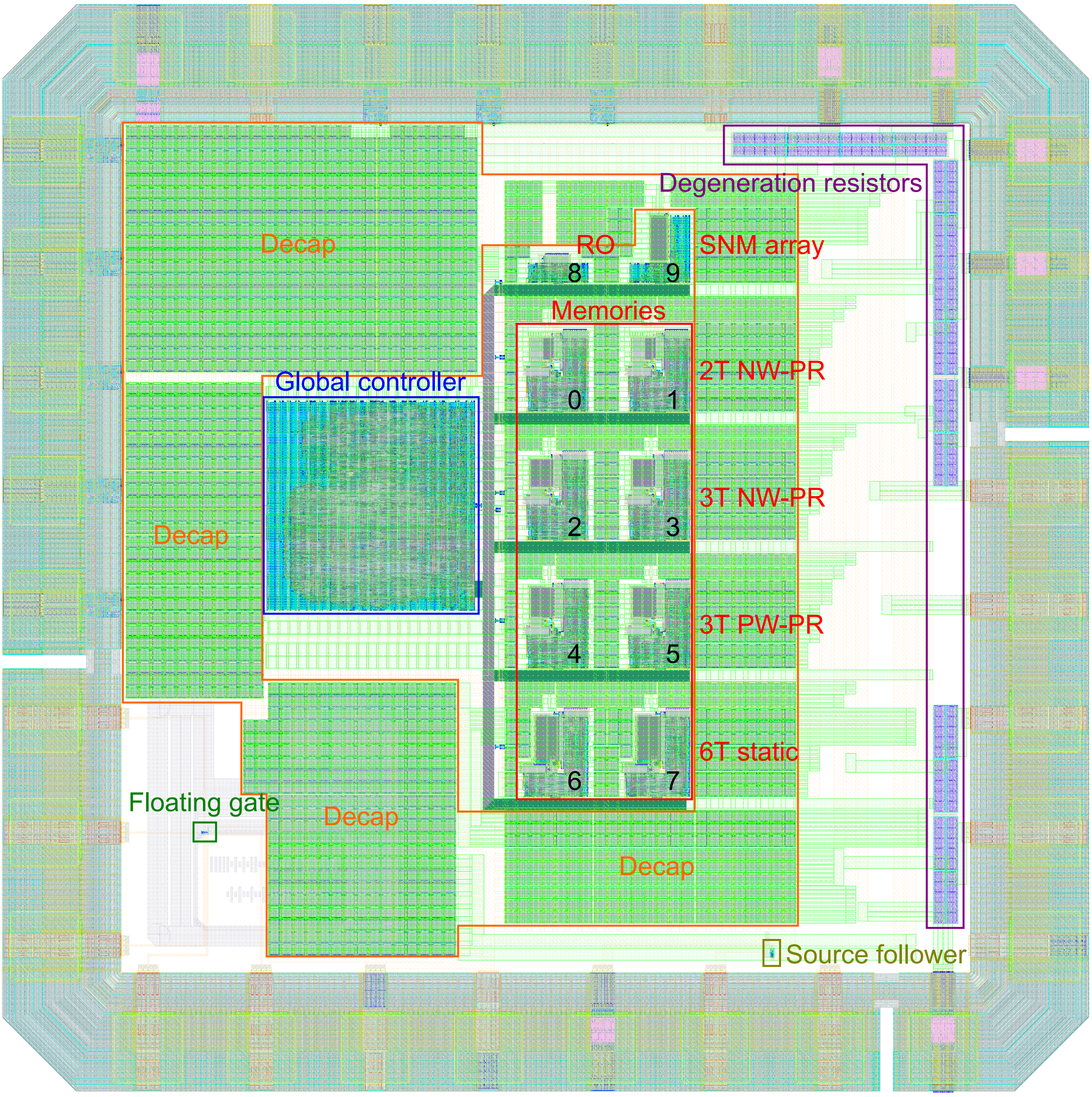


Figure 5.17: Top level layout with highlighted sub-circuits and numbered local controllers.

Delay chain characterisation
The approximate delay of the delay chains that are used throughout the design for a given setting can be measured with the following method.

1. Enable the RO by setting the eighth bit of the settings register of local controller number nine.
2. Measure the output frequency of the RO for at least two delay settings.
3. Estimate the delay of the delay chain using: $t_{\text{delay chain}} \approx \frac{1}{2 \cdot f_{\text{RO}}}$
4. Linearise the setting and delay pairs to find the offset (minimum delay) and resolution (delay step) of the delay chain.

Sense amplifier characterisation
The offset and noise of the sense amplifiers of a certain memory can be measured by using the following setup.

1. Enable sense amplifier test mode by setting the 26th bit of the settings register of the selected memory's local controller, and set reasonable read delays for the memory type.
2. Set the two reference voltage inputs.
3. Run a program that performs 1023 reads on the selected memory and accumulates the results in the error counters.
4. Repeat for various sense amplifier test reference voltages around the reference voltage.
5. Plotting the average read result for each sense amplifier over a range of input voltage differences will result in a figure similar to fig. 4.19, which shows the average result of 64 comparisons at various $V_{\text{SA,test}}$ voltages in a transient noise simulation. Fitting the result to the cumulative distribution function (CDF) of a normal distribution gives us the offset in the mean and the input-referred noise standard deviation in the standard deviation.

The previous method performs the measurements for all 32 sense amplifiers of the selected memory at the same time. The mean and standard deviation of the input-referred offset can be estimated from the distribution of the offsets of all sense amplifiers.

Read bitline curves
The read bitline curves that were extracted from simulation and used in the memory model can also be measured using the following method on a selected row of cells from a particular memory.

1. Run a program that does the following 1023 times:
   Write all-0 to the row of cells.
   Wait for a specified hold time.
   Read cell and accumulate the results.
2. Repeat the previous program for various reference voltage to find the point when the read result of a cell flips between 0 and 1. Using the sense amplifier statistics, the distribution of the bitline voltage for a specific cell and hold time can be found.
3. Repeat for all-1 data.
4. Repeat for various hold times.

Using the described method, a bitline curve can be measured for every cell. Depending on the range of hold times, either all cells or only a limited number of cells can be measured. At room temperature, the program will likely terminate in the order of $10\,\text{ms}$, while it may take several minutes at cryogenic temperatures for a hold time where the bitline margins start to drop significantly. From the room temperature characterisation, the worst cells can be selected for characterisation at cryogenic temperatures, assuming that the worst cells at room temperature are the worst cells at cryogenic temperatures. This assumption can also be verified by characterising some average and good room temperature cells at cryogenic temperatures. Finally, a statistical retention time can be calculated from the obtained cell statistics.

Read latency
The total read latency for a given set of timing settings can be measured by using the launch-capture hardware added to each memory.

1. Run a program that does the following:

    Write all-0 to a row and read it. Stop if the read is wrong.

    Write all-1 to a row and read it. Stop if the read is wrong.

    Decrement the capture delay chain setting and repeat.

2. Extract the critical setting from the program results and perform multiple write/read combinations around this setting to check if there is a statistical component to it.

With the previous method, the critical capture delay chain setting is obtained, including statistics about it. Using the delay chain characterisation results, this setting can be turned into a delay which gives the read latency of the memory. This can be repeated for various memory settings and used to find peripheral overhead on top of the read delay chain settings.

Read and write energy
The read and write energies per operation measurements are quite straightforward.

1. Run a program that continuously writes to/reads from a memory and measure the supply input current of the memory.
2. Compare measured input current with the input current when the memory is inactive to find the amount of current that is added due to the memory operations.
3. Multiply the additional input current with the supply voltage and divide by the number of operations per second to find the average energy needed for a write/read operation.

The previous method can be applied to each of the memories and compared with the values obtained from the memory model.

Static cell leakage power
The static cell leakage power can simply be measured by observing the supply current to the static arrays without applying any operations. We can obtain the leakage power by multiplying the current with the supply voltage. By varying the supply voltage, the leakage as a function of the applied voltage can be observed, which is expected to decrease exponentially. This may allow for efficient static memories at reduced supplies.

Static noise margin
Measuring the SNM of the static cells requires two measurement steps. In the first step, we characterise the source follower to find its input-to-output transfer curve. In the second step, we can cycle through the cells for various input voltages and measure the output voltage. Selection of the cells can be done through the manual bus controller. Using the input-to-output transfer curve of the source follower, the output voltages of the cells can be found from the measured source follower output.

Row hammer resilience
At cryogenic temperatures, row hammer disturbances in dynamic memories can severely limit the reliability due to increased retention time [14], [57]. The retention time increase allows more disturbances (read/write operations) between refreshes, the accumulation of which could cause a row hammer induced memory fault. This is mostly observed for commercial, single-transistor dynamic memories, but may also be the case for one of our dynamic memories.

This can be tested by observing the difference in time-to-failure between two situations. In the first case, data is written to a row and read after a certain inactive wait time. In the second case, read and/or write operations are applied on the rows neighbouring the row of interest during the wait time. If the reads after an equal wait time are significantly different, the memory is sensitive to row hammer. By varying the frequency of the operations on the neighbouring rows, a critical amount of operations between refreshes can be found which can cause too much degradation such that refreshes fail.

Floating gate device
Floating gate devices can be operated using various methods described in literature, all of which can be tested using the presented layout. The floating gate voltage could be influenced by Fowler-Nordheim tunnelling only [79], (impact ionized) hot carrier injection only [81], or a combination [74], [75], [80].

Since all methods require operating voltages outside the rated technology boundaries, these devices should be tested last, or using a separate chip sample. This prevents causing damage to a chip that is used for testing the memories, since it may lead to hard to trace or anomalous behaviour.

Increasing the floating-gate voltage using Fowler-Nordheim tunnelling can be tested as follows. First, the readout PMOS transistor is biased with a small current and a small resulting source-to-drain voltage. The voltage must be small to prevent hot carrier injection into the floating gate and should result in a constant output voltage. The readout PMOS transistor is in a source follower configuration and shows the floating gate voltage at the output plus one threshold voltage. Next, the tunnelling capacitor terminal is pulled high, which will also increase the floating gate voltage slightly due to coupling and therefore the output voltage. Due to Fowler-Nordheim tunnelling, charges should be tunnel onto the floating gate, resulting in a slow increase of the floating-gate voltage. When the tunnelling capacitor terminal is pulled down again, the output voltage should be higher than before the tunnelling. The rate of change of the floating-gate voltage is proportional to the tunnelling current.

The floating-gate voltage can be decreased by using Fowler-Nordheim tunnelling again, or by using hot carrier injection. By applying a large source-to-drain voltage across the readout PMOS transistor, hot holes are generated which can cause hot electrons on impact. These hot electrons can tunnel through the transistor gate dielectric and lower the floating-gate voltage again.

Apart from using a source-follower configuration of the readout PMOS transistor, its transfer characteristic can also be measured where its gate terminal is the gate coupling capacitor terminal. The charge on the floating gate will cause an apparent threshold voltage shift which can be observed in the transfer characteristic of the readout PMOS transistor.

5.4. Conclusion

To test the various memories with sufficient flexibility, a controller hierarchy is designed with a single global programmable controller and multiple local memory controllers, connected to each other through a bus. The global controller is connected to the chip pads such that it can be programmed through a shift register, and the memory designs are connected to the local controllers. The inputs and outputs of the SNM measurement structure and delay chain RO are also connected to chip pads to allow for SNM characterisation of the static cell design and characterisation of the delay chain. A floating gate device is also connected to the pads to allow for characterisation of such a device.

A suggested test plan has been shown with an overview of tests than can be performed. These tests can be used to verify the outputs from the model and obtain data that can be used by the model. Additional experiments can also be performed, such as measurement of the SNM of the static cell design, determining the row hammer resilience of the different dynamic cell designs, and characterising a floating gate device.

6 Conclusion and future work

The goal of this thesis was to determine the best memory cell design for the various memories found throughout the classical controller operating at $4.2\,\mathrm{K}$ in a quantum computer. Using a model, three different embedded dynamic memory cell designs and one static memory cell design have been compared over a range of application specifications. Additionally, eight memories have been designed and sent out for manufacturing. These can be characterised with the aim to verify and improve the memory model.

6.1. Conclusion

Three embedded dynamic cell designs (2T NW-PR, 3T NW-PR, and 3T PW-PR) and one static cell design (6T) have been compared for different applications in a classical controller for a quantum processor at both room temperature and $4.2\,\mathrm{K}$. At room temperature, memories in most applications can best be implemented using a static cell design, except for high frequency applications with more than 10^7 memory operations per second. At this point, the high refresh power of the dynamic cell designs is offset by the reduced operation power of the dynamic cell designs. At $4.2\,\mathrm{K}$, this boundary moves to 300 memory operations per second, making embedded dynamic cell memories a viable alternative at $4.2\,\mathrm{K}$. The choice of dynamic cell design depends on the ratio of read and write operations, with the 2T NW-PR cell design being preferred for applications with at most 1.77 reads per write on average, while the 3T NW-PR cell design is preferred for applications with more than 1.77 reads per write on average. This means that the 2T NW-PR cell design is better suited for queues and working memory in write-heavy algorithms, while the 3T NW-PR cell designs is preferred for lookup-tables and working memory in read-heavy algorithms.

To verify the model conclusions, eight memories have been designed for tape-out. For each cell design, two memories have been designed with different threshold-flavour devices to mitigate the threshold voltage increase at cryogenic temperatures. The eight memories are connected to a programmable microprocessor, which can run programs with at most 32 instructions, allowing for flexible testing and characterisation. Additionally, structures for measuring the static noise margin of the static cell design, characterisation of the delay chain design used in the timing circuits, and characterisation of a floating gate device are included.

6.2. Discussion

Since the reliability of the results of the model at $4.2\,\mathrm{K}$ is low, the correctness of the conclusions can only be verified by measurement of the memories. Especially the scaling of the retention time of the dynamic cells is only based on comparable results in literature, since scaling of the low activation energy leakage sources in the used technology is unknown. Additionally, the effect of the combined cryo-CMOS characteristics on the cell designs is only approximated and needs to be verified. Measurement of the memories will provide insight into the combined effect of the cryo-CMOS characteristics and a point of reference for the model.

In this work, only a single fixed architecture is assumed and alternative architectures may lead to different results. While the model gives an impression of the trade-offs that are involved on the cell design selection level, the result of changing the architecture and array sizes is not explored.

6.3. Future work

This thesis presented the development of a model for comparing memory cell designs at 4.2 K and the design of a chip intended for verification and refinement of the model. Suggestions for extending this work are listed below.

- First of all, the memories have to be characterised in order to close the loop with the model. By executing the test plan described in section 5.3, data can be obtained to tune the model parameters to verify that its results are reliable, both at room temperature and at 4.2 K.
- For the different cell designs, additional degrees of freedom can be used. For example, lowering the supply voltage of the static cell memory may increase its area in the memory landscape. This can easily be simulated and added to the model and again be verified by using the test chip.
- Apart from varying the cells, the peripherals may also be optimised for a specific use-case. Various peripheral designs could also be included in the model, and characterised on-chip, to improve the memory design for a chosen application.
- Apart from the cells and the peripherals, the memory architecture can also be optimised for a specific use-case, for example by resizing the array dimensions or using multiple banks. This requires additional peripherals that need to be added to the model and designed for verification.
- If all previously mentioned suggestions are followed, a comprehensive memory model can be made that is valid at 4.2 K. This model could be used to find the best memory architecture, peripheral, and cell design for a given application and its requirements.
- Finally, if the floating gate device can be tested successfully in this technology, it may also be used to develop various circuits for the classical controller as suggested in [75], [80].

A

Layout legend

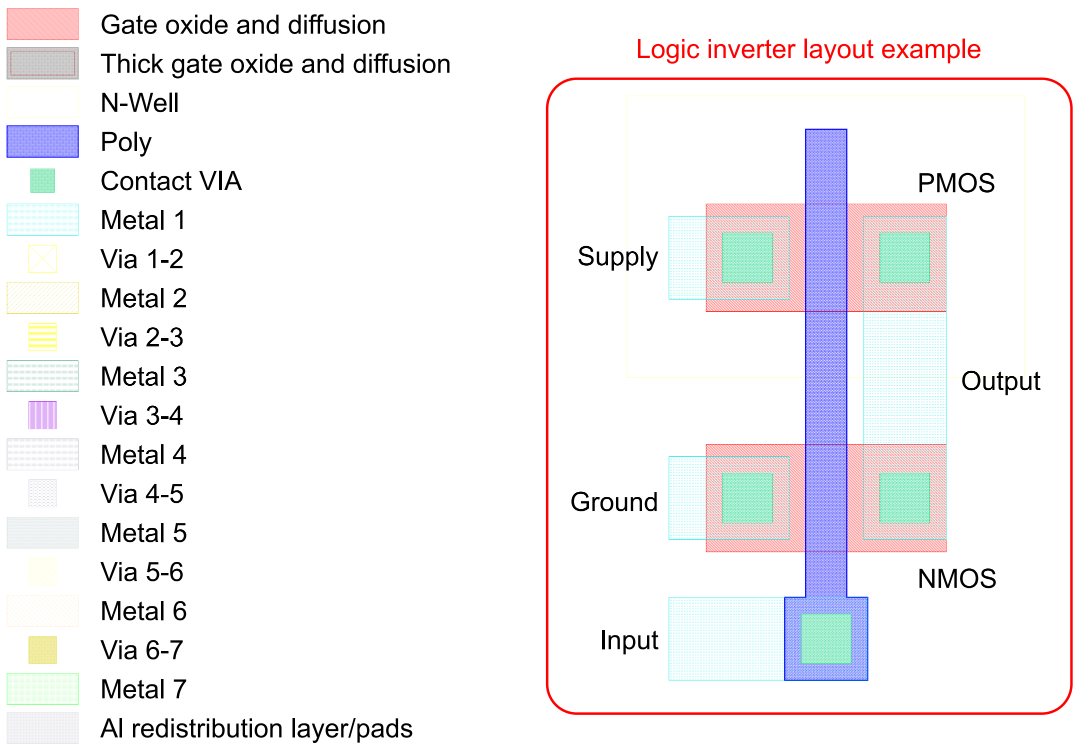


Figure A.1: Layer legend for layout figures.

Bibliography

[1] A. Montanaro, “Quantum algorithms: An overview,” *npj Quantum Information*, vol. 2, no. 1, pp. 1–8, 2016.

[2] A. W. Harrow and A. Montanaro, “Quantum computational supremacy,” *Nature*, vol. 549, no. 7671, pp. 203–209, 2017.

[3] C. Kloeffel and D. Loss, “Prospects for spin-based quantum computing in quantum dots,” *Annu. Rev. Condens. Matter Phys.*, vol. 4, no. 1, pp. 51–81, 2013.

[4] L. Vandersypen, H. Bluhm, J. Clarke, *et al.*, “Interfacing spin qubits in quantum dots and donors—hot, dense, and coherent,” *npj Quantum Information*, vol. 3, no. 1, pp. 1–10, 2017.

[5] G. Wendin, “Quantum information processing with superconducting circuits: a review,” *Reports on Progress in Physics*, vol. 80, no. 10, p. 106 001, 2017.

[6] M. Kjaergaard, M. E. Schwartz, J. Braumüller, *et al.*, “Superconducting qubits: Current state of play,” *Annual Review of Condensed Matter Physics*, vol. 11, pp. 369–395, 2020.

[7] E. Charbon, F. Sebastiano, A. Vladimirescu, *et al.*, “Cryo-CMOS for quantum computing,” in *2016 IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2016, pp. 13–5.

[8] F. Sebastiano, H. Homulle, B. Patra, *et al.*, “Cryo-CMOS electronic control for scalable quantum computing,” in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 1–6.

[9] T. Watson, S. Philips, E. Kawakami, *et al.*, “A programmable two-qubit quantum processor in silicon,” *nature*, vol. 555, no. 7698, pp. 633–637, 2018.

[10] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, “A quantum engineer’s guide to superconducting qubits,” *Applied Physics Reviews*, vol. 6, no. 2, p. 021 318, 2019.

[11] Y. Liu, L. Lang, Y. Chang, Y. Shan, X. Chen, and Y. Dong, “Cryogenic characteristics of multi-nanoscales field-effect transistors,” *IEEE Transactions on Electron Devices*, vol. 68, no. 2, pp. 456–463, 2020.

[12] S. S. Tannu, D. M. Carmean, and M. K. Qureshi, “Cryogenic-DRAM based memory system for scalable quantum computers: a feasibility study,” in *Proceedings of the International Symposium on Memory Systems*, 2017, pp. 189–195.

[13] F. Wang, T. Vogelsang, B. Haukness, and S. C. Magee, “DRAM retention at cryogenic temperatures,” in *2018 IEEE International Memory Workshop (IMW)*, IEEE, 2018, pp. 1–4.

[14] T. Kelly, P. Fernandez, T. Vogelsang, *et al.*, “Some Like It Cold: Initial Testing Results for Cryogenic Computing Components,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1182, 2019, p. 012 004.

[15] D. Min, I. Byun, G.-H. Lee, S. Na, and J. Kim, “Cryocache: A fast, large, and cost-effective cache architecture for cryogenic computing,” in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020, pp. 449–464.

[16] H. Seidl, M. Gutsche, U. Schroeder, *et al.*, “A fully integrated Al_2O_3 trench capacitor DRAM for sub-$100\,\mathrm{nm}$ technology,” in *Digest. International Electron Devices Meeting,*, IEEE, 2002, pp. 839–842.

[17] B. Keeth, R. J. Baker, B. Johnson, and F. Lin, *DRAM circuit design: fundamental and high-speed topics*. John Wiley & Sons, 2008, 2nd ed.

[18] X. Yuan, J.-E. Park, J. Wang, *et al.*, “Gate-induced-drain-leakage current in 45-nm CMOS technology,” *IEEE Transactions on Device and Materials Reliability*, vol. 8, no. 3, pp. 501–508, 2008.

[19] E. Garzón, Y. Greenblatt, O. Harel, M. Lanuzza, and A. Teman, “Gain-Cell Embedded DRAM Under Cryogenic Operation–A First Study,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2021.

[20] K. C. Chun, P. Jain, J. H. Lee, and C. H. Kim, “A 3T gain cell embedded DRAM utilizing preferential boosting for high density and low power on-die caches,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1495–1505, 2011.

[21] K. Ishibashi and K. Osada, *Low power and reliable SRAM memory cell and array design*. Springer Science & Business Media, 2011, vol. 31.

[22] X. Fu, L. Lao, K. Bertels, and C. G. Almudever, “A control microarchitecture for fault-tolerant quantum computing,” *Microprocessors and Microsystems*, vol. 70, pp. 21–30, 2019.

[23] X. Fu, L. Riesebos, L. Lao, *et al.*, “A heterogeneous quantum computer architecture,” in *Proceedings of the ACM International Conference on Computing Frontiers*, 2016, pp. 323–330.

[24] X. Fu, M. A. Rol, C. C. Bultink, *et al.*, “An experimental microarchitecture for a superconducting quantum processor,” in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, 2017, pp. 813–825.

[25] X. Fu, “Quantum Control Architecture: Bridging the Gap between Quantum Software and Hardware,” English, Ph.D. dissertation, Delft University of Technology, 2018, ISBN: 978-94-028-1305-0. DOI: `10.4233/uuid:8205cc34-30df-45f0-b6eb-8081bdb765b8`.

[26] A. Yadav, “CC-Spin: A Micro-architecture design for scalable control of Spin-Qubit Quantum Processor,” M.S. thesis, Delft University of Technology, Aug. 2019.

[27] P. Wang, X. Peng, W. Chakraborty, A. I. Khan, S. Datta, and S. Yu, “Cryogenic Benchmarks of Embedded Memory Technologies for Recurrent Neural Network based Quantum Error Correction,” in *2020 IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2020, pp. 38–5.

[28] R. W. Overwater, M. Babaie, and F. Sebastiano, “Neural-network decoders for quantum error correction using surface codes: A space exploration of the hardware cost-performance tradeoffs,” *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–19, 2022.

[29] J. P. G. Van Dijk, B. Patra, S. Subramanian, *et al.*, “A Scalable Cryo-CMOS Controller for the Wideband Frequency-Multiplexed Control of Spin Qubits and Transmons,” *IEEE Journal of Solid-State Circuits*, vol. 55, no. 11, pp. 2930–2946, 2020.

[30] R. M. Incandela, L. Song, H. Homulle, E. Charbon, A. Vladimirescu, and F. Sebastiano, “Characterization and compact modeling of nanometer CMOS transistors at deep-cryogenic temperatures,” *IEEE Journal of the Electron Devices Society*, vol. 6, pp. 996–1006, 2018.

[31] A. Beckers, F. Jazaeri, and C. Enz, “Characterization and modeling of 28-nm bulk cmos technology down to 4.2 k,” *IEEE Journal of the Electron Devices Society*, vol. 6, pp. 1007–1018, 2018.

[32] H. Homulle, “Cryogenic electronics for the read-out of quantum processors,” English, Ph.D. dissertation, Delft University of Technology, May 2019. DOI: `10.4233/uuid:e833f394-c8b1-46e2-86b8-da0c71559538`.

[33] P. ’t Hart, M. Babaie, E. Charbon, A. Vladimirescu, F. Sebastiano, *et al.*, “Characterization and modeling of mismatch in cryo-CMOS,” *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 263–273, 2020.

[34] Y. Zhang, T. Lu, W. Wang, *et al.*, “Characterization and Modeling of Native MOSFETs Down to 4.2 K,” *arXiv preprint arXiv:2104.03094*, 2021.

[35] A. Beckers, F. Jazaeri, A. Grill, S. Narasimhamoorthy, B. Parvais, and C. Enz, “Physical model of low-temperature to cryogenic threshold voltage in MOSFETs,” *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 780–788, 2020.

[36] F. Balestra and G. Ghibaudo, “Physics and performance of nanoscale semiconductor devices at cryogenic temperatures,” *Semiconductor Science and Technology*, vol. 32, no. 2, p. 023 002, 2017.

[37] B. Lengeler, “Semiconductor devices suitable for use in cryogenic environments,” *Cryogenics*, vol. 14, no. 8, pp. 439–447, 1974.

[38] M. Mehrpoo, B. Patra, J. Gong, *et al.*, “Benefits and challenges of designing cryogenic cmos rf circuits for quantum computers,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2019, pp. 1–5.

[39] H. Chiang, T. Chen, J. Wang, *et al.*, “Cold CMOS as a power-performance-reliability booster for advanced FinFETs,” in *2020 IEEE Symposium on VLSI Technology*, IEEE, 2020, pp. 1–2.

[40] Z. Wang, C. Cao, P. Yang, *et al.*, “Designing EDA-Compatible Cryogenic CMOS Platform for Quantum Computing Applications,” in *2021 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*, IEEE, 2021, pp. 1–3.

[41] R. Saligram, S. Datta, and A. Raychowdhury, “Scaled back end of line interconnects at cryogenic temperatures,” *IEEE Electron Device Letters*, vol. 42, no. 11, pp. 1674–1677, 2021.

[42] L. Deferm, E. Simoen, B. Dierickx, and C. Claeys, “Anomalous latch-up behaviour of CMOS at liquid helium temperatures,” *Cryogenics*, vol. 30, no. 12, pp. 1051–1055, 1990.

[43] M. Deen, C. Chan, and N. Fong, “Operational characteristics of a cmos microprocessor system at cryogenic temperatures,” *Cryogenics*, vol. 28, no. 5, pp. 336–338, 1988.

[44] N. Yoshikawa, T. Tomida, M. Tokuda, *et al.*, “Characterization of 4 K CMOS devices and circuits for hybrid Josephson-CMOS systems,” *IEEE Transactions on Applied Superconductivity*, vol. 15, no. 2, pp. 267–271, 2005.

[45] R. Saligram, W. Chakraborty, N. Cao, Y. Cao, S. Datta, and A. Raychowdhury, “Power performance analysis of digital standard cells for 28 nm bulk CMOS at cryogenic temperature using BSIM models,” *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 7, no. 2, pp. 193–200, 2021.

[46] W. Link and H. May, “Low temperature characteristics of MOS single-transistor memory cells,” *Archiv Elektronik und Uebertragungstechnik*, vol. 33, pp. 229–235, 1979.

[47] F. Ware, L. Gopalakrishnan, E. Linstadt, *et al.*, “Do superconducting processors really need cryogenic memories? The case for cold DRAM,” in *Proceedings of the International Symposium on Memory Systems*, 2017, pp. 183–188.

[48] K. Kuwabara, H. Jin, Y. Yamanashi, and N. Yoshikawa, “Design and implementation of 64-kb CMOS static RAMs for Josephson-CMOS hybrid memories,” *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, pp. 1 700 704–1 700 704, 2012.

[49] G. Konno, Y. Yamanashi, and N. Yoshikawa, “Fully functional operation of low-power 64-kb Josephson-CMOS hybrid memories,” *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–7, 2016.

[50] M. Tanaka, M. Suzuki, G. Konno, Y. Ito, A. Fujimaki, and N. Yoshikawa, “Josephson-CMOS hybrid memory with nanocryotrons,” *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–4, 2016.

[51] R. C. Jaeger and T. N. Blalock, “Quasi-static RAM design for high performance operation at liquid nitrogen temperature,” *Cryogenics*, vol. 30, no. 12, pp. 1030–1035, 1990.

[52] W. H. Henkels, D.-S. Wen, R. Mohler, *et al.*, “A 4-Mb low-temperature DRAM,” *IEEE Journal of Solid-State Circuits*, vol. 26, no. 11, pp. 1519–1529, 1991.

[53] Q. Liu, T. Van Duzer, X. Meng, *et al.*, “Simulation and measurements on a 64-kbit hybrid Josephson-CMOS memory,” *IEEE Transactions on Applied Superconductivity*, vol. 15, no. 2, pp. 415–418, 2005.

[54] P. Wyns, R. Anderson, and W. Des Jardins, “Temperature dependence of required refresh time in dynamic random access memories,” in *Proceedings of the Symposium on Low Temperature Electronics and High Temperature Superconductors*, Electrochemical Society, vol. 88, 1988, p. 123.

[55] A. Weber, A. Birner, and W. Krautschneider, “Data retention analysis on individual cells of 256Mb DRAM in 110nm technology,” in *Proceedings of 35th European Solid-State Device Research Conference, 2005. ESSDERC 2005.*, IEEE, 2005, pp. 185–188.

[56] R. Saligram, S. Datta, and A. Raychowdhury, “CryoMem: A 4K-300K 1.3 GHz eDRAM Macro with Hybrid 2T-Gain-Cell in a 28nm Logic Process for Cryogenic Applications,” in *2021 IEEE Custom Integrated Circuits Conference (CICC)*, IEEE, 2021, pp. 1–2.

[57] G.-h. Lee, S. Na, I. Byun, D. Min, and J. Kim, “CryoGuard: A Near Refresh-Free Robust DRAM Design for Cryogenic Computing,” *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pp. 637–650, 2021.

[58] P. Wyns and R. L. Anderson, “Low-temperature operation of silicon dynamic random-access memories,” *IEEE Transactions on Electron Devices*, vol. 36, no. 8, pp. 1423–1428, 1989.

[59] J. A. Halderman, S. D. Schoen, N. Heninger, *et al.*, “Lest we remember: cold-boot attacks on encryption keys,” *Communications of the ACM*, vol. 52, no. 5, pp. 91–98, 2009.

[60] W. H. Henkels, N. C. Lu, W. Hwang, *et al.*, “A 12-ns low-temperature DRAM,” *IEEE Transactions on Electron Devices*, vol. 36, no. 8, pp. 1414–1422, 1989.

[61] G.-h. Lee, D. Min, I. Byun, and J. Kim, “Cryogenic computer architecture modeling with memory-side case studies,” in *Proceedings of the 46th International Symposium on Computer Architecture*, 2019, pp. 774–787.

[62] T. I. Chappell, S. E. Schuster, B. A. Chappell, *et al.*, “A 3.5 ns/77 K and 6.2 ns/300 K 64 K CMOS RAM with ECL interfaces,” *IEEE Journal of Solid-State Circuits*, vol. 24, no. 4, pp. 859–868, 1989.

[63] V. P.-H. Hu and C.-J. Liu, “Static Noise Margin Analysis for Cryo-CMOS SRAM Cell,” in *2021 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT)*, IEEE, 2021, pp. 1–2.

[64] H. Hanamura, M. Aoki, T. Masuhara, O. Minato, Y. Sakai, and T. Hayashida, “Operation of bulk CMOS devices at very low temperatures,” *IEEE Journal of Solid-State Circuits*, vol. 21, no. 3, pp. 484–490, 1986.

[65] T. Van Duzer, L. Zheng, S. R. Whiteley, *et al.*, “64-kb hybrid Josephson-CMOS 4 Kelvin RAM with 400 ps access time and 12 mW read power,” *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, pp. 1 700 504–1 700 504, 2012.

[66] L. Geck, A. Kruth, H. Bluhm, S. van Waasen, and S. Heinen, “Control electronics for semiconductor spin qubits,” *Quantum science and technology*, vol. 5, no. 1, p. 015 004, 2019.

[67] R. Kalla, B. Sinharoy, W. J. Starke, and M. Floyd, “Power7: IBM’s next-generation server processor,” *IEEE micro*, vol. 30, no. 2, pp. 7–15, 2010.

[68] T. J. O’Gorman, “The effect of cosmic rays on the soft error rate of a DRAM at ground level,” *IEEE Transactions on Electron Devices*, vol. 41, no. 4, pp. 553–557, 1994.

[69] L. Bautista-Gomez, F. Zyulkyarov, O. Unsal, and S. McIntosh-Smith, “Unprotected Computing: A Large-Scale Study of DRAM Raw Error Rate on a Supercomputer,” in *SC’16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2016, pp. 645–655.

[70] G. Asadi and M. B. Tahoori, “Soft error rate estimation and mitigation for SRAM-based FPGAs,” in *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*, 2005, pp. 149–160.

[71] J.-L. Autran, S. Serre, S. Semikh, D. Munteanu, G. Gasiot, and P. Roche, “Soft-error rate induced by thermal and low energy neutrons in 40 nm SRAMs,” *IEEE Transactions on Nuclear Science*, vol. 59, no. 6, pp. 2658–2665, 2012.

[72] M. Yao, M. F. Cabanas-Holmen, and E. H. Cannon, “Direct Measurement Structure of SRAM SNM,” The Boeing Company Huntington Beach United States, Tech. Rep., 2019.

[73] *AMBA AXI and ACE Protocol Specification Version H.c*, Arm Limited, Cambridge, England, 2021.

[74] J. Hasler, S. Kim, and F. Adil, “Scaling floating-gate devices predicting behavior for programmable and configurable circuits and systems,” *Journal of Low Power Electronics and Applications*, vol. 6, no. 3, p. 13, 2016.

[75] M. Castriotta, E. Prati, and G. Ferrari, “Floating-gate transistor at cryogenic temperature: Characterization and modelling of tunnelling and hot electrons injection,” in *2020 IEEE Silicon Nanoelectronics Workshop (SNW)*, IEEE, 2020, pp. 89–90.

[76] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, “A precision CMOS amplifier using floating-gate transistors for offset cancellation,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 2, pp. 280–291, 2007.

[77] E. Ozalevli, H.-J. Lo, and P. E. Hasler, “Binary-weighted digital-to-analog converter design using floating-gate voltage references,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 4, pp. 990–998, 2008.

[78] V. Srinivasan, G. Serrano, C. M. Twigg, and P. Hasler, “A floating-gate-based programmable CMOS reference,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 11, pp. 3448–3456, 2008.

[79] S. Millner, “Development of a multi-compartment neuron model emulation,” Ph.D. dissertation, Heidelberg University, 2012.

[80] J. Hasler, N. Dick, K. Das, B. Degnan, A. Moini, and D. Reilly, “Cryogenic Floating-Gate CMOS Circuits for Quantum Control,” *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–10, 2021.

[81] T. Rizzo, S. Strangio, and G. Iannaccone, “Time Domain Analog Neuromorphic Engine Based on High-Density Non-Volatile Memory in Single-Poly CMOS,” *IEEE Access*, vol. 10, pp. 49 154–49 166, 2022.