Stony Brook University

Energy-Efficient Local Storage Design With Superconductor Reciprocal Quantum Logic

A Dissertation Presented

by

Zuoting Chen

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

December 2015

Stony Brook University

The Graduate School

Zuoting Chen

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Mikhail Dorojevets – Dissertation Advisor
Associate Professor, Department of Electrical and Computer Engineering

Sangjin Hong – Chairperson of Defense
Professor, Department of Electrical and Computer Engineering

Emre Salman
Assistant Professor, Department of Electrical and Computer Engineering

Jennifer L. Wong
Assistant Professor, Department of Computer Science
Stony Brook University

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Energy-Efficient Local Storage Design With Superconductor Reciprocal Quantum Logic

by

Zuoting Chen

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

2015

Superconductor single flux quantum (SFQ) technology is one of the promising candidates for energy-efficient high-performance computing. A new generation of SFQ logics, Reciprocal Quantum Logic (RQL) with no static power dissipation in bias resistors, offers an opportunity to dramatically decrease energy consumption in superconductor processors. Although several low complexity RQL processing units have already been demonstrated, the use of RQL for local storage design has not been explored yet.

The objective of this dissertation is to design on-chip local storage structures such as memory, register files, and caches with RQL technology and analyze their energy efficiency, complexity, and performance characteristics. The physical chip design of these RQL

storage units is not feasible at this point because both CAD tools for physical VLSI chip design and as well as a target fabrication process are under development at Northrop-Grumman Systems Corp. (Baltimore, MD) and MIT Lincoln Laboratory, respectively. In order to achieve our goal, the layout-aware cell-level design process using VHDL RQL cell library developed at the Ultra High Speed Computing Lab in Stony Brook University has been used. The SBU VHDL RQL cell library specifies the dynamic and stand-by energy consumption, latency, JJ complexity, and approximate sizes of individual cells based on the input received from the JJ-level RQL designers. Clock propagation skew and wire delays are accounted for during circuit simulation. The circuit simulation is done with Mentor Graphics design and verification tools. As a result of the work, key characteristics of the 8.5 GHz multiported RQL storage structures with their capacity in the range of 1-4 kbit have been determined. The average energy consumption of the RQL storage designs is ~3.0-9.5 fJ/bit/operation at room temperature and the cryo-cooling efficiency is 0.1%.

The data from this dissertation also reveal the critical issues that need to be considered in the RQL storage design. These will be helpful for further development on superconductor VLSI design.

Contents

List of Figures

List of Tables

Acknowledgments

This may be the most difficult section I need to write in this dissertation. Memories of happiness and frustration come back to me as I am trying to remember all the people I would like to acknowledge. I wish I do not leave anyone out. But for those I may have overlooked, I truly appreciate and enjoy our time together.

First and foremost I'd like to thank Prof. Mikhail Dorojevets, my advisor at Stony Brook University. His hands-on guidance, unrelenting conviction and strive for perfection has sharpened my skills and made this dissertation possible. Besides my research, Prof. Dorojevets always gives me a lot of advice on my academic progress and career. I am extremely grateful for that.

This dissertation could not be complete without the help of the other members in my committee: Prof. Sangjin Hong, Prof. Emre Salman and Prof. Jennifer Wong. I am very grateful for their feedback, patience and encouragement. Also, I thank Rachel Ingrassia and Susan Hayden of the Electrical and Computer Engineering department for helping me work through the labyrinth of paperwork and graduation process.

No research can be finished without the help of fellow colleagues. In the Ultra High Speed Computing Laboratory, I am very lucky to work with Dr. Christopher Ayala. He is a very nice and skillful person. He guided me into the research on Reciprocal Quantum Logic and helped me with all questions in the research and career. I cannot forget the time with Hao Chen, with whom I have completed the multiplier project in the lab. He also gave me a lot of information during my job hunting. I am looking forward to meet you in California in December. Thanks to Artur Kasperek, I wish you and your family all the best in Poland.

I would like to thank Prof. Alex Doboli and Cristian Ferent, who provided me a chance to work on analog circuit layout at the Mixed-Domain Embedded Systems Laboratory in my master's. To Qilin Miao, I had a great time with you in the lab, and you help me a lot when I was busy with computer architecture project and algorithm exams.

There are countless people I am grateful to in my undergraduate at Sun

Yat-sen University. As a freshman, I didn't know what to do and how to plan my future until Prof. Xiao Huang and Yanxiang Bao introduced me into the world of electrical design. After that, I decided to start my career as an electrical engineer. I met lots of friends during my study: Zongdie Li, Jinsong Mao, Kechao Huang and Chunhua Zhou. We worked together and learnt from each other. That was the most enjoyable time in my life.

The friendships I have forged during my graduate studies at Stony Brook University are very dear to me. I value the friendships of other fellow Ph.D students, Wenxiang Ding, Yang Liu, Jie Zhao, Fanshu Jiao, Qi Fan, Yifan Hu, Zongyuan Liu, Yutong Pang, Xinyu Zhang, Lei Wang, Huafeng Huang, etc. I thank each and every one of them for their outpouring support and wonderful memories.

And finally, I would like to thank my family and my girlfriend. To my parents, Lucheng Chen and Qiongjin Mo, thank you for your unending, nurturing love. To my grandfather Yongxi Chen and my Uncle Guoping Mo, thank you for your affection and education that have made me the person I am today. To my girlfriend, Yaqun Yu, thank you so much for standing beside me and support me during my busiest time.

Zuoting Chen
Saint James, New York USA
November 2015

Vita

Zuoting Chen was born in Guangdong, China on February 7, 1989. He finished his B.S.E.E. program at Sun Yat-sen University, Guangdong, China in June 2011. In the same year, he was admitted into the M.S.E.E. program at Stony Brook University. In December 2012, he graduated from M.S.E.E. program and admitted into the Ph.D. program in Computer Engineering with a minor area in Circuits and VLSI at Stony Brook University. He passed the Computer Engineering Qualifying Exam in April 2013. Since July 2012, he has been working as a research assistant under Dr. Mikhail Dorojevets, conducting studies on superconducting Reciprocal Quantum Logic storage and processing units. He is a graduate student member of IEEE.

Publications

The following is a list of publications that are a result of the research conducted for this dissertation:

1. M. Dorojevets, **Z. Chen**, C. L. Ayala, and A. K. Kasperek, "Towards 32-bit Energy-Efficient Superconductor RQL Processors: The Cell-Level Design and Analysis of Key Processing and On-Chip Storage Units," *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 3, Jun. 2015.
2. M. Dorojevets, **Z. Chen**, "Fast pipelined storage for high-performance energy-efficient computing with superconductor technology," in *IEEE Proc. Emerging Technologies for a Smarter World (CEWIT), 2015 12th International Conference Expo on*, pp.1-6, 19-20 Oct. 2015.

Chapter 1

Fundamentals of Superconductor Technology

Outline

1.1 Introduction

High-performance and energy-efficiency are the critical issues in modern digital integrated circuit design. Superconductor technology, with high operation speed and zero resistance at DC, has become one of the promising candidate for high-performance and energy-efficient computing. The essential element in the superconductor technology is a Josephson junction (JJ). A JJ is built

according to the Josephson effect that was predicted by Brian David Josephson in 1962 [1, 2, 3] and experimentally confirmed in Bell Labs. In late 1960s, Josephson junction (JJ) was proved to be a suitable choice for fast circuit design. Without the need of doping materials, the fabrication technology for JJ-based integrated circuit was simpler than the fabrication technology for semiconductor [4]. The further improvements in the fabrication technology made the Josephson junction integrated circuits easier to fabricate. As demonstrated, some superconductor logic gates can operate at 770 GHz [5]. To provide a cryogenic environment for the superconductor circuit, a cryostat device is required to keep the circuit at the temperature of ~4.2 K. Even with the unavoidable high costs of cryo-cooling, superconductor technology has advantages over CMOS in terms of energy efficiency [6].

The remainder of this chapter provides an introduction of the superconductor technology family. A brief review of the prior work in this area is also given. In Chapter 2, we will focus on the concept of Reciprocal Quantum Logic (RQL), which is used in this research. A tunable RQL cell library and its target fabrication process are also included. Chapter 3 reviews the current status and challenges of the research on superconductor storage, as well as the opportunity of designing local storage units with RQL.

Four types of local storage units are studied in this research: 1) RAM with 1 read and 1 write ports in Chapter 4, 2) register file with 2 read and 1 write ports in Chapter 5, 3) write-through and write-back cache in Chapter 6, and 4) first-in first-out buffer in Chapter 7.

Finally, Chapter 8 makes some final discussions and conclusions of this research, follow by a brief outlook on further research in superconductor RQL local storage and processors.

1.2 Josephson Junctions

A Josephson junction is a two-terminal device with two niobium superconductors separated by a thin aluminum oxide barrier, as shown in Figure 1.1 on page 3. The barrier is ~1-nm-thick so that both normal electrons and Cooper pairs can tunnel through [7, 8, 9]. Only Cooper pairs can tunnel when zero voltage is applied to a JJ; both Cooper pairs and normal electrons can tunnel when a voltage is applied across the junction. The fundamental relation between the phase drop $\phi = \varphi_1 - \varphi_2$ and voltage drop $V = \mu_1 - \mu_2$ in a JJ can be shown in Equation 1.1, where h is Planck's constant and e is the electron charge [10].

$$\frac{d\phi}{dt} = (\frac{2e}{h})V(t) \tag{1.1}$$

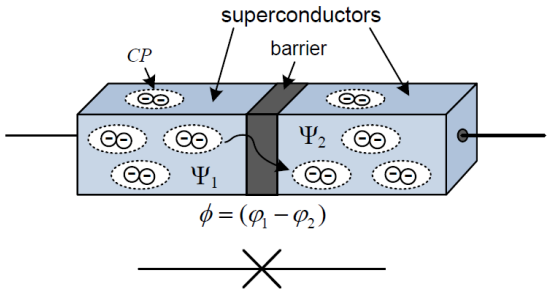


Figure 1.1: Josephson junction structure and circuit symbol [15]. © 2011 IEEE.

The flow of Copper pairs (supercurrent) in a JJ can be determined by Equation 1.2, where ϕ is the phase drop and I_c is the critical current of a JJ that depend on its area and barrier transparency [10].

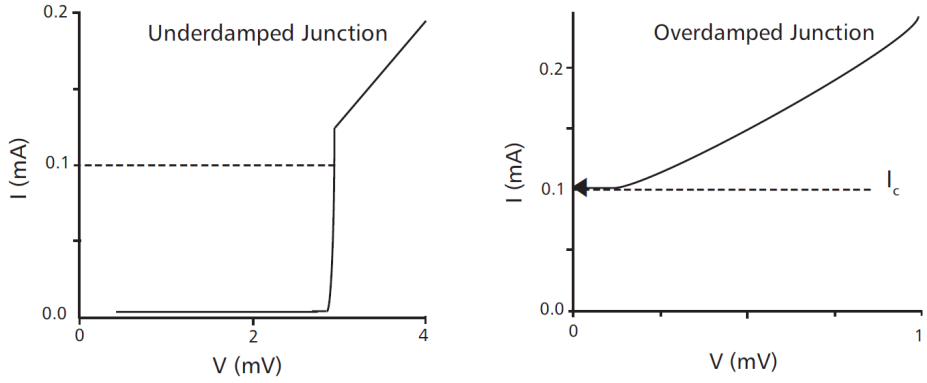
$$I_s = I_c \sin \phi \tag{1.2}$$

For large signals, another three components should be taken in to account, as in Equation 1.3, where C is the junction capacitance and R is its "normal resistance" [10]. $I_f(t)$ is the current noise in R, which is very small.

$$I(t) \approx I_c \sin \phi + C\frac{dV}{dt} + \frac{V}{R} + I_f(t) \tag{1.3}$$

A JJ can operate in two different modes: voltage-state (latching) and single flux quantum (SFQ). Figure 1.2 on page 4 shows the I-V characteristics of these two modes[11, 12, 13]. In the first mode, as shown in Figure 1.2a on page 4, the junction switches from $V = 0mV$ to $V = V_g$ when the current exceeds I_c. The junction can be reset to zero-voltage stage when the current is reduced near zero. This two-state logic voltage approach was the first JJ-based superconductor family and very popular for superconductor computing projects in IBM and Japan during 1970s and 1980s. In order to reset the junction to zero-voltage, this logic uses by AC power system. However, the demonstrations had shown that the speed of this technology was limited to about 1 GHz. This was the reason why IBM and other research groups stopped working with the latching logic technology in 1980s [14].

A Josephson junction can also operate in SFQ mode if the junction is shunted by a small resistor. The I-V curve is shown in Figure 1.2b on page 4. If the current is higher than I_c, the voltage can increase continuously. This approach is the fundamental of the Rapid Single Flux Quantum (RSFQ) logic, which will be discuss in the next section.



(a) Voltage-state latching JJs. **(b)** SFQ-based non-latching JJs.

Figure 1.2: I-V characteristics of Josephson junction [11].

1.3 Rapid Single Flux Quantum Logic

Rapid Single Flux Quantum logic has been developed to increase the operation speed of JJ-based circuits. RSFQ circuits are based on JJs shunted by a resistor which allows JJ working at the SFQ mode. Moreover, RSFQ circuits are DC powered instead of AC powered in voltage-state logic. RSFQ circuit had been demonstrated at clock frequency of 100 GHz in 1980s.

RSFQ uses the natural superconductor property that the current in a closed superconducting ring is the multiples of the magnetic flux quantum Φ_0. RSFQ is based on the manipulation and transportation of these magnetic flux quanta [16]. The logical states '1' and '0' are represented by a presence and absence of a single flux quantum, respectively. Figure 1.3 on page 5 shows a simplest SFQ circuit, Superconducting QUantum Interference Device (SQUID), that is used to store the digital information. The SQUID is a superconductor ring interrupted by a JJ. The JJ plays a role of I/O interface by manipulating flux quanta inside the superconducting ring. If the current level is lower than the critical current I_c, the JJ allows a constant superconducting current (dc) to pass through without voltage drop and the phase difference between two superconductors remains constant. If the current exceeds I_c, a voltage across the junction is developed and a very short voltage pulse $V(t)$ (so called SFQ pulse) is generated. The SFQ pulse is used to carry digital information and can travel through the circuit. Logical '1' is represented by the arrival of a SFQ pulse, and logical '0' is represented by the absence of SFQ pulses. Equation 1.4 shows the relationship between SFQ pulse and magnetic flux quantum, where h is Planck's constant and e is the electron charge.

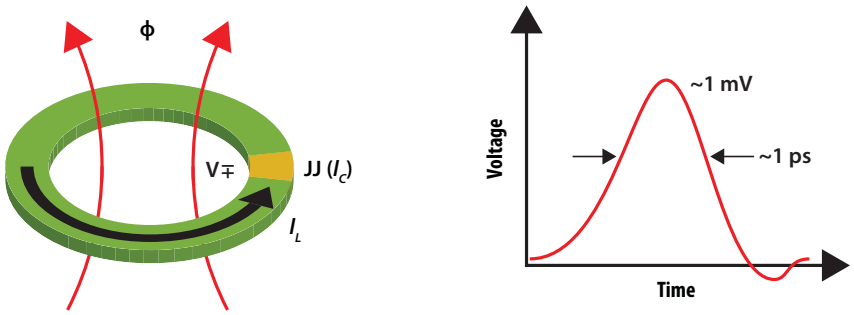


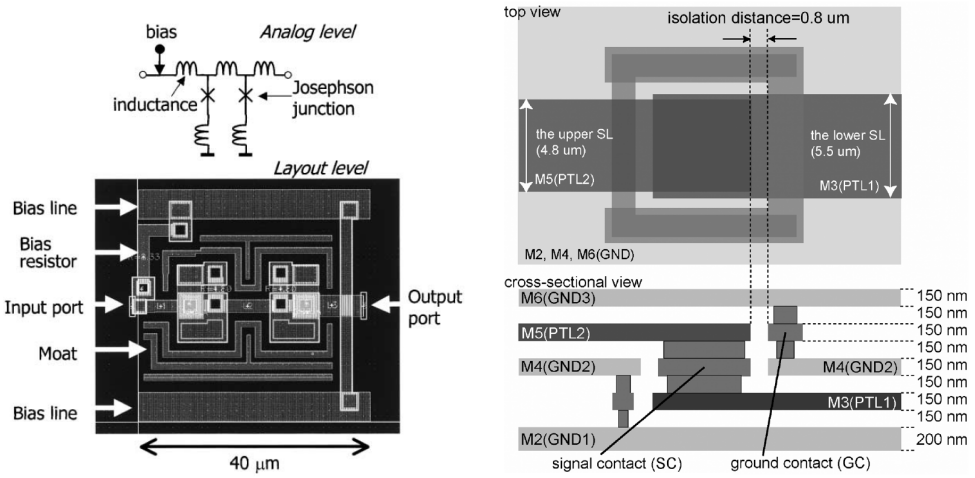
Figure 1.3: Superconductor ring and SFQ pulse [16].

$$\int V(t)dt = \Phi_0 = \frac{h}{2e} \approx 2.07 \text{ mV} \times \text{ps} \tag{1.4}$$

SFQ pulses can be transferred through two types of connections: 1) active Josephson junction transmission lines (JTL) and 2) passive transmission lines (PTL). Figure 1.4a on page 6 shows the simplest JTL. PTLs provide fast connections for long distance propagation of SFQ pulses at a velocity of ~100 µm/ps [17, 18, 19]. Active JJ-based drivers (TX) and receivers (RX) are required to amplify the pulse before sending over a PTL, and re-amplify it from the PTL, respectively. This is a propagation delay overhead for benefiting from the high-speed and low-loss transmission [20]. A driver-PTL-receiver schematic is shown in Figure 1.4c on page 6. As shown in Figure 1.4b on page 6, the ISTEC 10 kA/cm^2 advanced process supports two PTL layers [17].

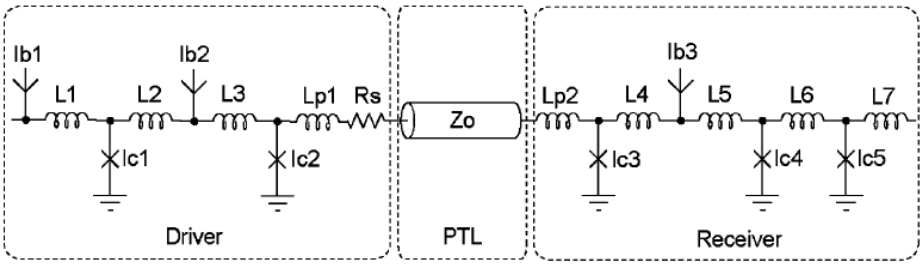
A RSFQ D flip-flop is shown in Figure 1.5a on page 7. A storage loop is formed by an input junction J1, an inductor and a junction J3. J1 is bias at $0.7I_c$. When an input SFQ pulse arrives at data input, the current in J1 will exceed the critical current, and J1 will switch and store one flux quantum into the loop. This flux quantum adds a current in J3. The arrival of a clock pulse will switch J3, generating an output SFQ pulse and resetting loop back to zero-state. If there is no flux quantum in the loop, the clock pulse cannot switch J3 because the current in J3 is insufficient. As the result, no SFQ pulse is generated at the output. Figure 1.5b on page 7 shows the RSFQ D flip-flop waveform.

Power dissipation in RSFQ has two parts: dynamic power and static power. The dynamic power comes from the energy loss during a junction switch, and the static power comes from the bias current distribution network that uses bias resistors to distribute the bias current to JJs. The major contribution of the power dissipation is the static power, which is ~99% of the total power in some large-scale designs [22]. The total power of RSFQ circuit can be



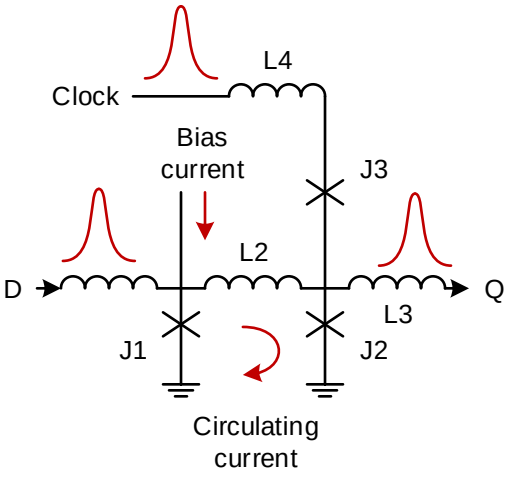
(a) JTL schematic and layout.

(b) PTL via connection.

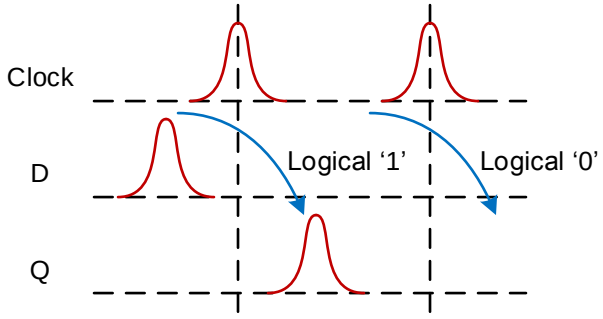


(c) Driver-PTL-Receiver schematic.

Figure 1.4: JTL and PTL connections used in superconductor logic [17, 21]. © 2009 IEEE.



(a) Circuit schematic.



(b) Circuit waveform.

Figure 1.5: RSFQ D flip-flop [1, 11, 15, 20].

estimated by Equation 1.5 at 4.2 K, where I_b is the total bias current and V_b is the bias voltage which is 2.6 mV for the HYPRES 4.5 kA/cm^2 process, and 2.5 mV for the ISTEC 10 kA/cm^2 process. To estimate the power consumption at room temperature, we need to take into account the power to cool the circuit down to 4.2 K, as shown in Equation 1.6, where $1000W_{room}/W_{cryo}$ is the cryostat efficiency [20].

$$P_{cryo} = I_b V_b \tag{1.5}$$

$$P_{room} = P_{cryo}\left(1000\frac{\mathrm{W}_{room}}{\mathrm{W}_{cryo}}\right) \tag{1.6}$$

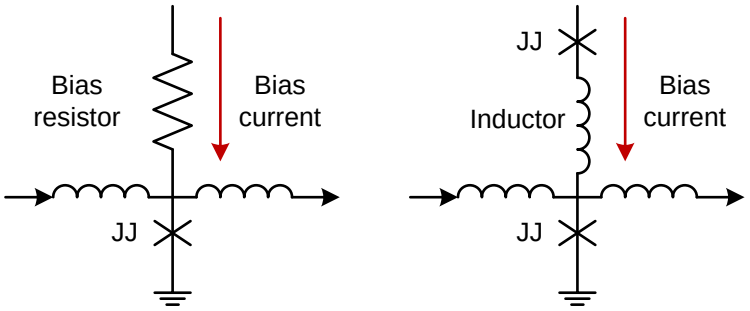
RSFQ logic has some serious weaknesses. One of them is the high static power consumption. As we mentioned above, the static power dissipated on bias resistors is significantly higher than the dynamic power. This can be a problem as the circuit complexity grows, which makes RSFQ unsuitable for energy-efficient VLSI design. To overcome this problem, several new logic families are developed to reduce the static power.

1.4 Energy-Efficient RFSQ Logic

Energy-efficient Rapid Single Flux Quantum (ERFSQ) logic is developed to reduce the static power consumption [22, 23]. Unlike RSFQ that uses bias resistors to supply a bias current, ERFSQ uses JJs (act like a natural current limiter) to distribute current to the logic gates, as shown in Figure 1.6 on page 9. To achieve this, the voltage on the power line must be equal or greater than the maximum possible DC voltage used to power the circuit [20]. This DC voltage is determined by the clock frequency, as shown in Equation 1.7, wheref_c is the circuit clock frequency and $\Phi_0 \approx 2.07$ mV $\times$ ps [23]. Equation 1.8 shows the ERSFQ power. Unlike the fixed 2.5-2.6 mV bias voltage in RSFQ, bias voltage in ERFSQ is in a range of 20-100 µV for clock frequencies in a range of 10-50 GHz.

$$V_b = \Phi_0 f_c \tag{1.7}$$

$$P = I_b V_b = I_b \Phi_0 f_c \tag{1.8}$$



(a) RSFQ schematic. **(b)** ERFSQ schematic.

Figure 1.6: Biasing in RSFQ and ERSFQ circuits [11].

1.5 Brief Introduction to Other Superconductor Logic Families

Besides ERFSQ, designer also developed some other SFQ logic families to eliminate the static power consumption in RSFQ. There are: 1) low-voltage SFQ [24], 2) reciprocal quantum logic (RQL) [25, 26, 27, 28, 29, 30, 31], and 3) adiabatic quantum-flux-parametron (AQFP) [32]. In this section, we will have a brief introduction to these new logic families.

Low-voltage SFQ reduces the static power by reducing the bias resistor in the bias current distribution network. The total power consumption is reduced by 93% compared to RFSQ. Simply reducing the bias resistor would decrease of the bias current, causing deterioration of the circuit operation and interaction between circuits. An LR-loading technique [33] is used to stabilize the circuit operation. With this approach, the static power consumption is reduced to the same level as the dynamic power consumption.

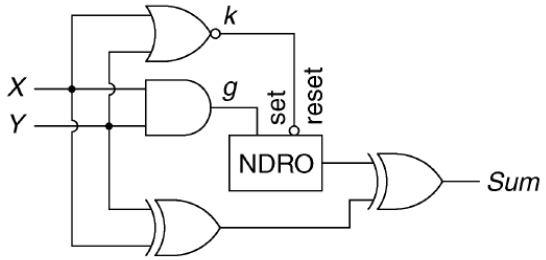
Reciprocal Quantum Logic is another new technology aims to reduce static power consumption. RQL uses AC power lines to drive the circuit, which also sever as a clock. RQL circuits with zero static power consumption have been demonstrated . This is the technology I am going to study in this research. More information will be in the next chapter.

AQFP is developed to reduce the dynamic power. It is an AC powered logic based on the quantum flux parametron (QFP) [34, 35, 36], which has the advantages of high-gain, high speed and high robustness. In AQFP, the QFP gates are working in adiabatic mode, which reduce the dynamic energy consumption to its fundamental limit.

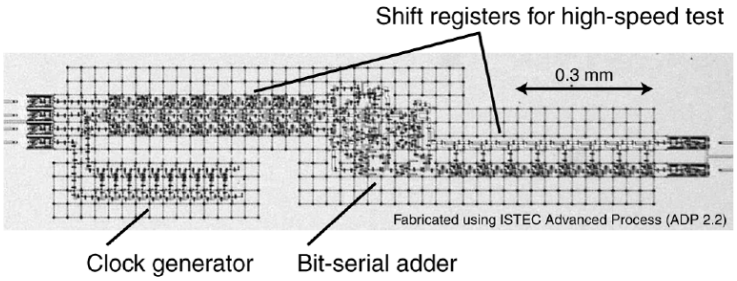
1.6 Prior Work on SFQ Logic

1.6.1 100 GHz SFQ Bit-Serial Adder

A bit-serial RSFQ adder was developed by a reasearch group from Nagoya University, Yokohama National University, the Superconductivity Research Laboratory-ISTEC, and Kyoto University in 2011 [37]. This adder is targeted at the frequency of 100 GHz. Because multiple SFQ pulse cannot travel in the loop path, a concept of state transitions is adopted. A non-destructive readout (NDRO) gate is used to save the carry, as shown in Figure 1.7a on page 10. This circuit is based on the ISTEC 10 kA/cm^2 fabrication process and it has a sufficient DC bias margin of ±18% at frequencies of up to 60 GHz. The correctness has been verified at the frequency of up to 93 GHz.



(a) Bit-serial schematic.



(b) Microphotograph of a bit-serial adder test circuit.

Figure 1.7: 100 GHz bit-serial adder [37]. © 2011 IEEE.

1.6.2 Fujitsu's 8-bit DSP Microprocessor

Fujitsu Laboratories designed an 8-bit DSP microprocessor as one of the early approach to build practical chip using JJ technology in 1990 [14, 38]. This design is based on latching logic and has 23,000 JJs in a 5 mm × 5 mm chip. This DSP includes a 13-bit 16-function ALU [39], an 8-bit × 8-bit multiplier,

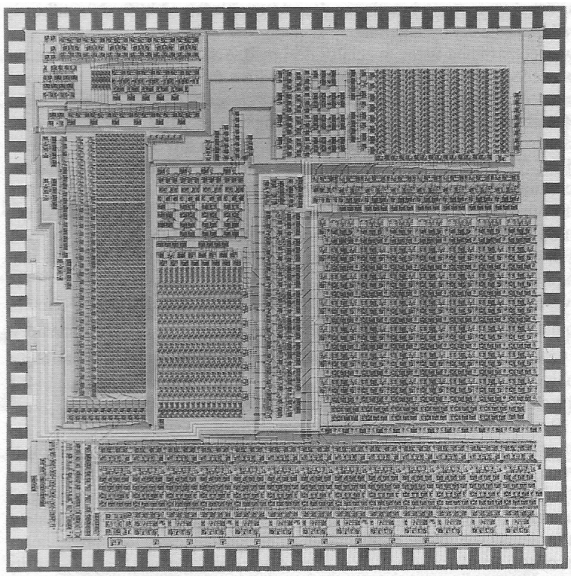


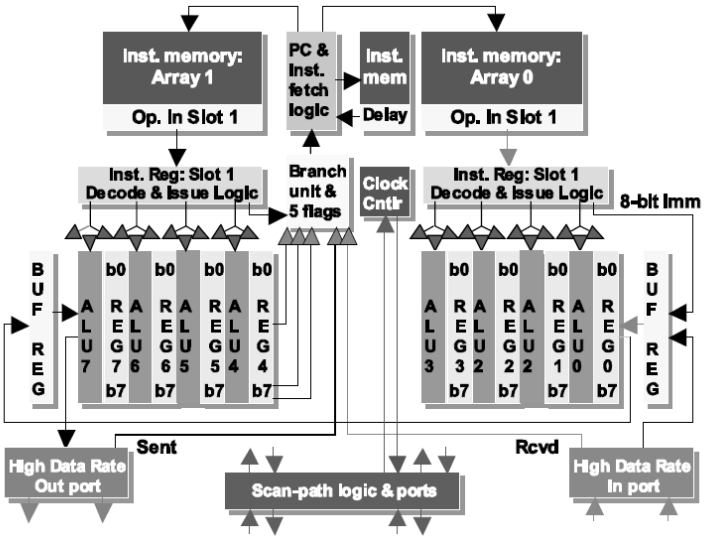
Figure 1.8: Microphotograph of Fujitsu's 8-bit DSP based on latching logic [14]. © 1992 IEEE.

a 64 word × 24-bit instruction ROM, a 16 word × 8-bit coefficient ROM, and two 16 word × 8-bit data RAM. The access time of the instruction ROM is 200 ps, and that of the RAM cells is 130 ps. The estimated frequency is up to 1 GHz and the power is 12 mW. At that time, it was about 100 times faster and one-tenth the power of conventional CMOS DSPs. All functions in this chip have been successfully demonstrated.

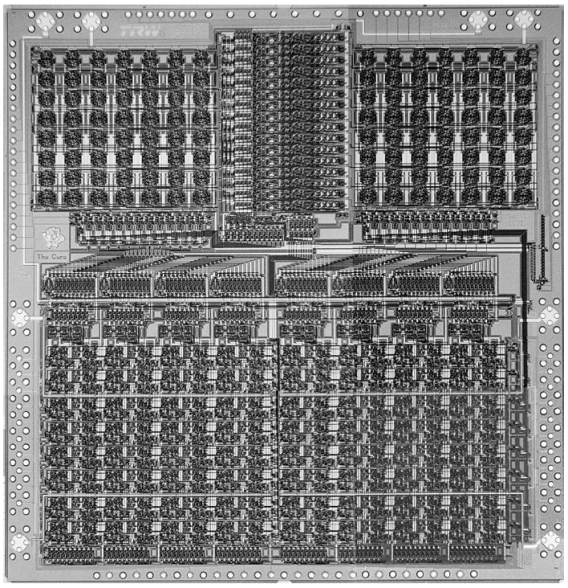
1.6.3 FLUX-1 Microprocessor

The FLUX-1 microprocessor was developed by Stony Brook University (SBU) and TRW (now Northrop Grumman) with the target frequency of 17-20 GHz [11, 40, 41, 42]. The goal was to study and understand the architectural and design challenges for 20+ GHz RSFQ processors. This 8-bit RSFQ microprocessor employs partitioned synchronous dual-op Long-Instruction-Word (LIW) architecture. The last version of the processor, FLUX-1R, was fabricated in TRW's 4 kA/cm^2, 1.75 μm Josephson junction technology in 2002. This chip has 63,107 Josephson junctions on a 10.35 mm × 10.65 mm^2 die and the energy consumption is ~9.5 mW at 4.2 K. The block diagram and the physical layout are shown in Figure 1.9 on page 12.

Only a low-capacity instruction memory was implemented in this design. The memory has the following features:



(a) Block diagram.



(b) Microphotograph of the second chip.

Figure 1.9: The FLUX-1 8-bit RSFQ microprocessor [40]. © 2003 IEEE.

- There are 2 16-row × 16-bit arrays in the memory.
- Each arrays issues one instruction to the instruction register per cycle.
- Wave pipelining is used in the memory to ensure one dual-op instruction per cycle.
- The complexity of the memory is ~16,500 JJs, and takes over 25% of the total JJs complexity (not including PC and instruction fetch circuit).

1.6.4 CORE1 Microprocessor

The CORE1 is a joint project by Nagoya University, and Yokohama National University, and Communications Research Laboratory at Kobe, and International Superconductivity Technology Center (ISTEC) Superconductor Research Lab (SRL) at Tsukuba [43, 44, 45, 46].

This project started with a simple processor known as CORE1α which was fully demostrated in 2003. It has the following features [43]:

- A 32 byte shift-register-based memory for instructions and data.
- Two 8-bit data registers and a bit-serial ALU.
- Seven 8-bit instructions in the instruction set.
- Non-pipelined processing and control logic.
- 1 GHz system clock and 16-21 GHz local clock.
- ~7,220 JJs on a 3.4 × 3.2 mm^2 die.
- Power consumption of 2.3 mW at 4.2 K.

In 2007, a more advanced version, CORE1β was demonstrated with the following features [44]:

- A 16-bit instruction register (IR).
- Four 8-bit register file.
- Two cascaded bit-serial ALUs.
- ~9,498 JJs.
- Power consumption of 3.0 mW at 4.2 K.

The final version, CORE1γ , is implemented with cache memory and pipeline techniques [46](Figure 1.10 on page 14).

- 16 byte and 8 byte shift-register-based direct-mapped instruction cache and data cache, respectively.

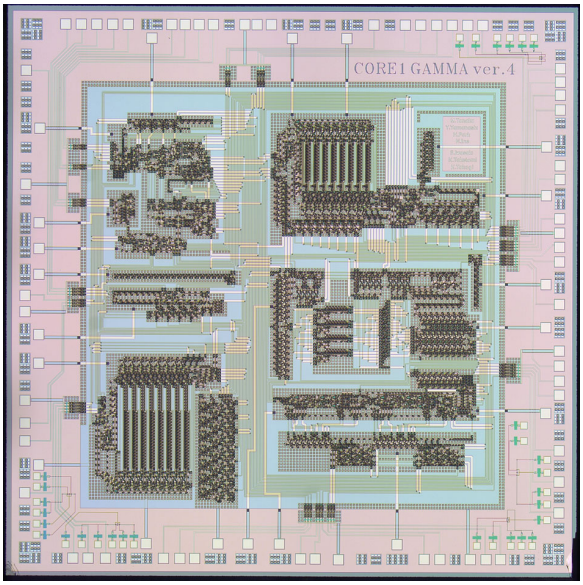


Figure 1.10: Microphotograph of the CORE1γ 8 $\times$ 8 mm^2 chip [46].

- Eight-stage pipeline.
- 22,302 JJs on 6.36 $\times$ 6.36mm^2 area on an 8 $\times$ 8 mm^2 die.
- Estimated power consumption of 6.56 mW at 4.2 K.

1.6.5 20 GHz 8-bit RSFQ Frontier Datapath

An 8-bit RSFQ datapath has been developed in a joint project between SBU and HYPRES. This 8-bit datapath is based on a 32-bit Frontier data-flow microarchitecture developed at SBU [47]. The complete cell-level datapath design and its verification have been done by the SBU team using a SBU VHDL cell library tuned to the HYPRES 1.5 μm 4.5 kA/cm^2 fabrication process. The HYPRES team developed the physical layout design, fabricated and tested two datapath components, namely an asynchronous wave-pipelined ALU [48, 49] and an 8 $\times$ 8-bit multi-port register file [50]. This NDRO-based register file can perform two simultaneous read operations and one write operation.

The ALU has been demonstrated at the target frequency of 20 GHz with $\pm$5% DC bias margins and the register file has been demonstrated at a low frequency with $\pm$4% DC bias margins.

Chapter 2

Reciprocal Quantum Logic

Outline

2.1 Overview

Reciprocal Quantum Logic (RQL) [25, 26, 27, 28, 29, 31] is a new generation of superconductor SFQ logic that is currently considered as one of the solutions for high-performance energy-efficient computing. In this chapter, we will first discuss the fundamental of RQL, and the finished work with RQL. After that,

the SBU RQL cell library and fabrication process in Massachusetts Institute of Technology Lincoln Laboratory (MIT LL) that is used in this research will be provided.

Unlike RSFQ, RQL is powered by AC power lines inductively coupled with the cells in series, as shown in Figure 2.1 on page 17. This AC power line is also serves as a clock to the circuit [25]. Without bias resistors, RQL have approximately two orders of magnitude better energy efficiency than similar superconductor circuits implemented with the older Rapid Single Flux Quantum (RSFQ) logic.

Figure 2.2 on page 17 shows how data are encoded in RQL. Logical '1' is represented by a pair of SFQ pules: a positive pulse during the positive half cycle follow by a negative pulse during the negative half cycle. The positive pulse is used to set the SQUID to '1' state and then propagate forward, while the negative pulse is used to reset the SQUID to '0' state, preventing signal from traveling backward [27]. Because the circuit can be reset by the following negative pulse, the trailing reset wave that is necessary in RSFQ is not required in RQL. This simplifies the gate design and creates a combinational logic behavior similar to CMOS.

The dynamic power consumption in RQL circuit is defined as the number of junction switch when a pair of positive and negative pulses arrives. As shown in Equation 2.1, where n is the number of switched JJs, $I_c(i)$ is the junction's critical current, and f is the frequency.

$$P_{dynamic} \approx \frac{2}{3} \sum_{i=1}^{n} I_c(i) \Phi_0 f \tag{2.1}$$

In addition to the dynamic energy, RQL circuits have very small power consumption due to the parasitic coupling of clock lines to the JJ shunt resistors of the inactive cells in a stand-by mode of operation with no input pulses received during a cycle time [29]. We call it standby power consumption to distinguish it from the static power consumption in bias resistors of RSFQ cells. The accuracy of the energy model used in our simulation tools is expected to be within 10%.

In contrast to CMOS, the interconnect energy consumption does not depend on wire lengths because of the ballistic propagation of pulses with the speed of ~0.1 mm/ps over superconducting Nb PTLs with negligible dielectric losses. Energy losses in passive microwave components are not considered in our study.

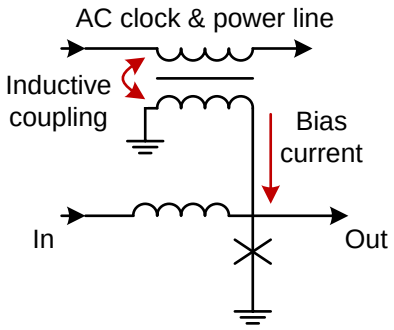


Figure 2.1: RQL schematic.

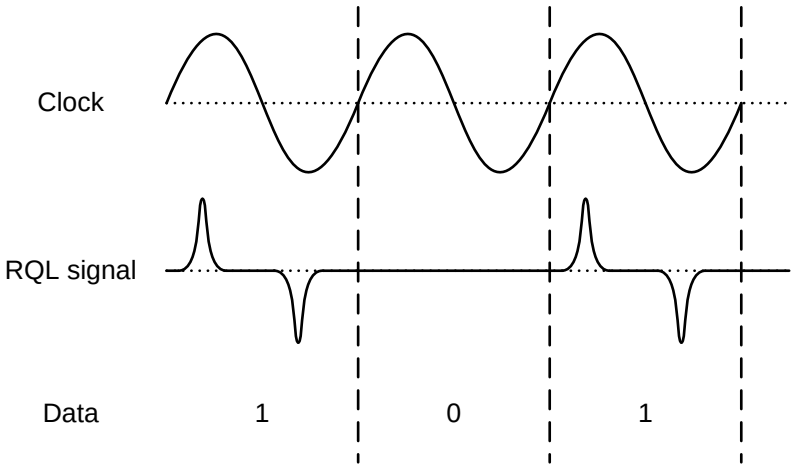
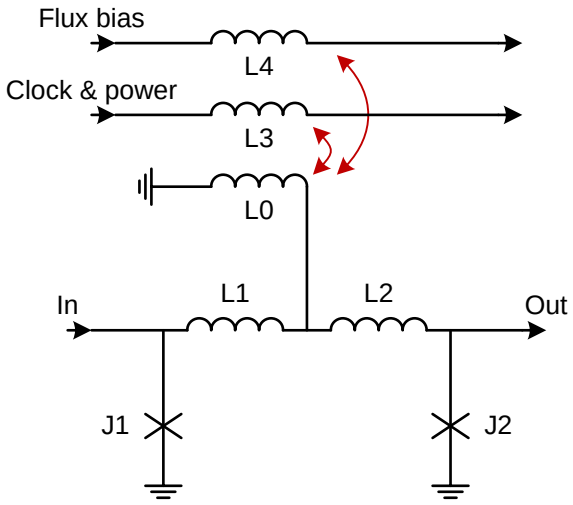


Figure 2.2: Data in RQL.

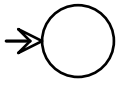
2.2 Josephson Transmission Line in RQL

A RQL JTL is formed by a series of connection cells. A connection cell is an inductive loop formed by junctions J1 and J2 and inductors L1 and L2, as shown in Figure 2.3 on page 18. L0 is coupled to L3 in the AC power line to provide bias current through the junctions. The AC power line provides positive bias current during the positive half of the clock cycle and negative bias current during the negative half of the clock cycle, which is sufficient for positive and negative SFQ pulses to switch the junctions, respectively. An additional DC flux is used to induce a flux of $\Phi_0/2$ in the connection cell, so the cell can switch between two symmetric states: $+\Phi_0$ and $-\Phi_0$.

The connection cell can be used to amplify the pulse energy. This is achieved by stepping up the critical current of the cell from one to the next. The connection cell in JTL can amplify SFQ pulse energy by stepping up the critical current from one cell to the next. By increasing the critical current at a factor of $\sqrt{2}$, the SFQ pulse energy is doubled [27]. With this feature, JTL can be used as a SFQ splitter with fan-out of 2.



(a) Circuit schematic.



(b) block diagram symbol.

Figure 2.3: RQL connection cell [27].

2.3 Four-Phase Clocking

A SFQ pulse can propagate through the connection cells as long as the bias current is sufficient [27]. During the negative half-cycle, the SFQ pulse that propagates forward in the positive half-cycle can move backward. To resolve this issue, RQL uses four-phase clocking: two clock lines with the phase difference of $\pi/2$. A connection cell can be coupled with one of the clock line in

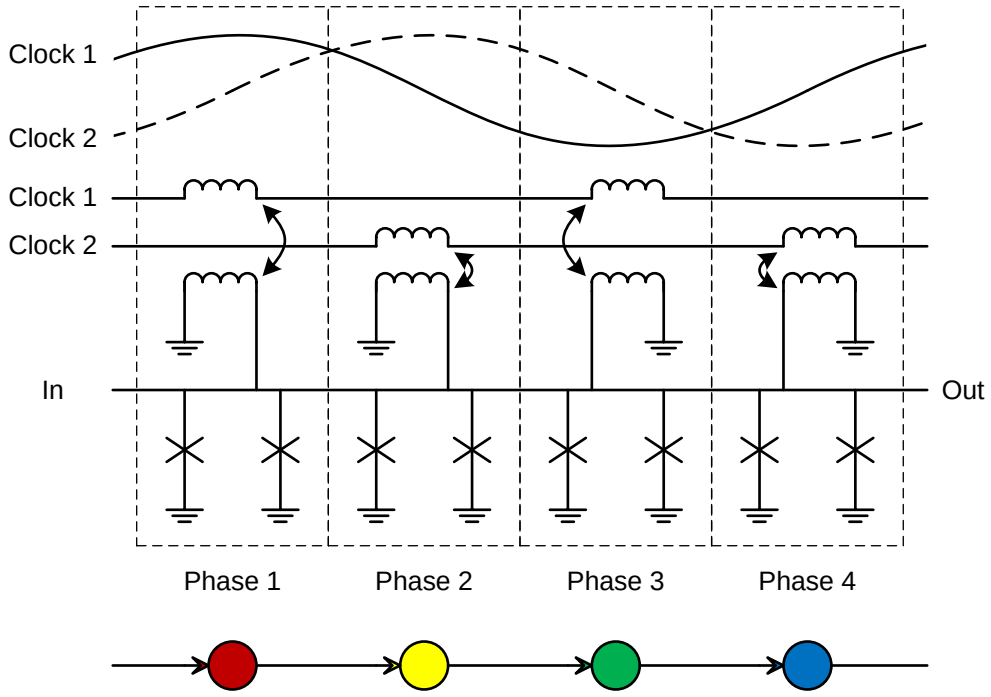


Figure 2.4: Four-phase clocking in a RQL one-cycle delay register [27].

a wound or counter wound faction. This produces a total of four phases with the difference of 0, $\pi/2$, π, and $3\pi/2$ as shown in Figure 2.4 on page 19. When one phase is reaching the end of the phase, the next phase has already started in the next connection cell, allowing pulse propagation to continue. [27]

The SFQ pulse is self-synchronized in four-phase clocking. When a pulse reach a phase boundary (where the next connection cell is clocked by next phase), the pulse can wait for the start of the next phase and continue propagation. With this feature, pulses with different arrival times can be synchronized at the phase boundary without additional synchronize circuit.

In addition, several connection cells can be clocked with the same phase, as shown in Figure 2.5 on page 20. The SFQ pulse can travel through the connection cells in the same phase until it reaches the end of the phase's timing window [27]. The number of connection cells that can be set in one phase is determined by the phase time (clock period / 4) and the propagation delay through the connection cells. Total propagation delay through the connection cells cannot exceed the phase time.

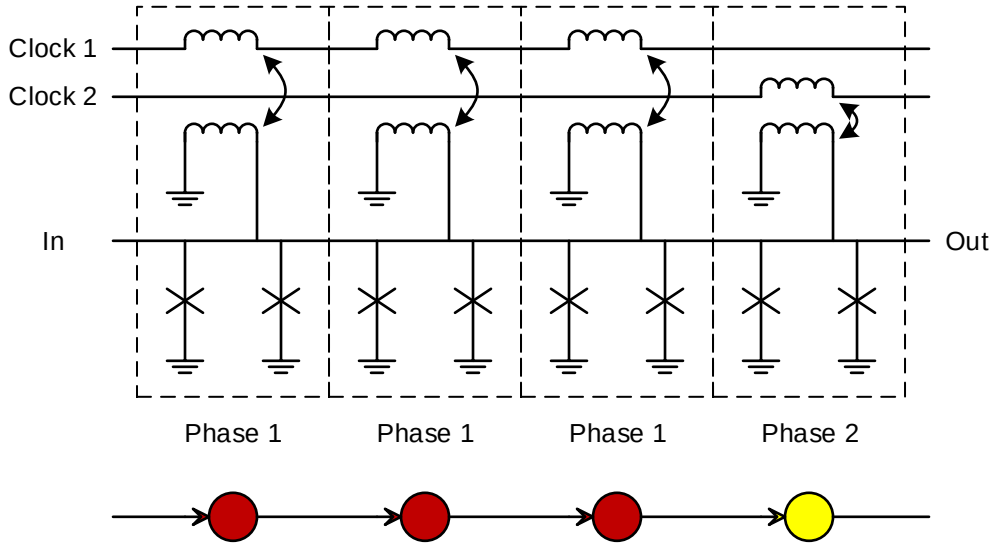


Figure 2.5: Several RQL cells in a same phase [27].

2.4 Other Key RQL Cells

2.4.1 Passive Transmission Line Receiver

All superconductor circuits use lossless passive transmission lines (PTLs) to transmit SFQ pulses (without any capacitive charging involved) over long distance connections. As shown in Figure 2.6 on page 20, a connection cell (acting as a driver) sends a SFQ pulse to PTL. A PTL receiver receives and re-amplifies the pulse from the PTL. Like a connection cell, a PTL receiver should be inductively coupled with the clock & power line and set to one of the four phases in a clock cycle. To be safely received by a PTL receiver, a pulse from a passive transmission line needs to arrive during the PTL reception window, which is in the center of a phase and the width is one eighth of a clock cycle (half phase).

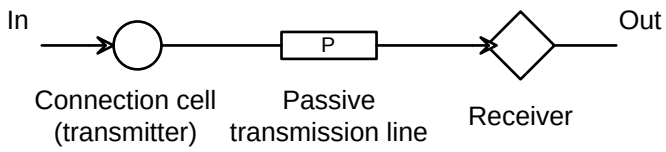
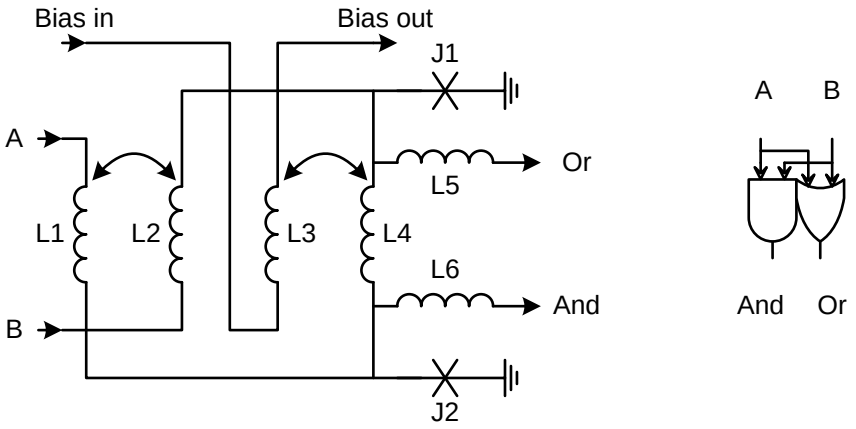


Figure 2.6: Receiver and passive transmission line.

2.4.2 AndOr Gate

The schematic of an AndOr gate is shown in Figure 2.7a on page 21, J1 at the OR output is preferentially biased by $\Phi_0/2$. This junction will switch when the first pulse arrives at either input. After switching, a pulse is generated at Or output and the flux state of the gate is reversed, which means J2 is preferentially biased. If the second pulse arrives, J2 will switch and a pulse will be generated at And output. If two pulses arrive simultaneously, J1 and J2 switch and generate Or and And output at the same time. The AndOr gate symbol is in Figure 2.7b on page 21.



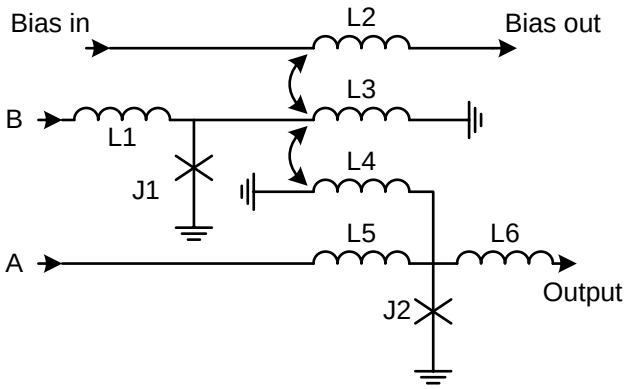
(a) AndOr gate schematic. **(b)** AndOr gate symbol.

Figure 2.7: AndOr gate [27].

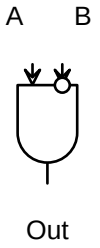
2.4.3 AnotB Gate

Figure 2.8a on page 22 shows the schematic of an AnotB gate. A pulse arrives at B switches J1, creating a negative current that block J2 from switching. This is so call output-inhibiting state, as a pulse at A cannot propagate to the output. This state will be cancelled out by the following reciprocal pulse after half cycle. If there is no input pulse at B, pulse arrives at A can switch J2 and generate output. There is a timing requirement for the AnotB gate: pulse at B should arrive earlier or simultaneously with pulse at A. The AnotB gate symbol is in Figure 2.8b on page 22.

Both AndOr and AnotB gates are unclocked. The bias current is supplied by the connection cells that drive the input SFQ pulse to or receive the output pulse from it. As the result, these gates can be place either in one phase or in



(a) AnotB gate schematic.



(b) AnotB gate symbol.

Figure 2.8: AnotB gate [27].

the boundary of two phases. The timing of the gate depends on the connection cell that supplies bias current to it.

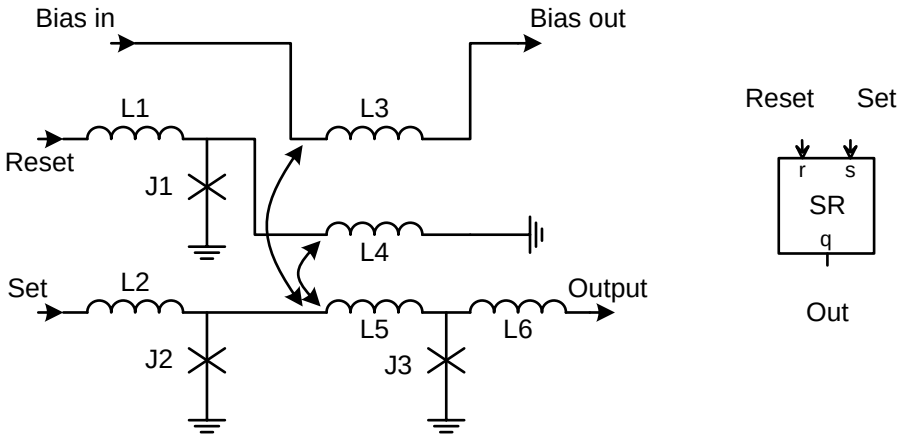
2.4.4 Set/Reset Gate

The Set/Reset gate is used to build the basic storage device with non-destructive readout in RQL. Figure 2.9a on page 23 shows the schematic of a Set/Reset gate. J2, J3 and L5 form a set memory loop, and J1 and L4 form a reset loop. This gate has an internal state that can be switch between two bi-stable flux states, which represent logical '1' and '0'.

Initially, the internal flux of the gate is $+\Phi_0/2$. If a pair of pulses arrives at Set port, the positive pulse applies $+\Phi_0$ to the gate and the internal flux is $+3\Phi_0/2$. Then J3 switches and generates a positive output and the internal flux is return to $+\Phi_0/2$. The following reciprocal pulse changes the internal flux to $-\Phi_0/2$. Any further pairs of pulses will change the internal flux to $+\Phi_0/2$ and back without output. We can define this as state '1' [27].

A pair of pulses that arrives at Reset can change the state of the gate. The leading positive pulse applies $-\Phi_0$ to the gate and changes the internal flux to $-3\Phi_0/2$. A negative output is generated and the internal flux is return to $-\Phi_0/2$. The reciprocal pulse changes the internal flux to $+\Phi_0/2$. The internal flux keeps switching between $-\Phi_0/2$ and $+\Phi_0/2$ for following reset pulse pairs until a pair of pulses arrives at Set input. We can define this as state '0' [27].

The Set/Reset gate symbol is shown in Figure 2.9b on page 23.



(a) Set/Reset gate schematic.

(b) Set/Reset gate symbol.

Figure 2.9: Set/Reset gate [27].

In order to allow negative output to propagate, the reset pulse should arrive at the negative clock cycle. As the result, the gate should operate in two phases, where Set input and output are in the same phase and Reset input is in another.

2.4.5 Non-Destructive Read-out Single-Bit Storage Cell

The Set/Reset gate only generates output when the internal flux state is changed. To preserve the output from the Set-Reset gate, Non-Destructive Read-Out (NDRO) single-bit storage cell is developed to preserve the output from the Set-Reset gate for multiple read operations, as shown in Figure 2.10 on page 24. The positive pulse (cause by set operation in the Set/Reset gate) from the Set/Reset gate can be stored in the connection cell until a negative reciprocal pulse (cause by reset operation in the Set/Reset gate) arrives. If there is a positive pulse stored in the connection cell, the AnotB gate is in output-inhibiting state. The pulses from read input will not generate an output. In other words, a logical '0' is read. If there is no positive pulse stored in the connection cell, pulses from read input can propagate through the AnotB gate. In this case, a logical '1' is read.

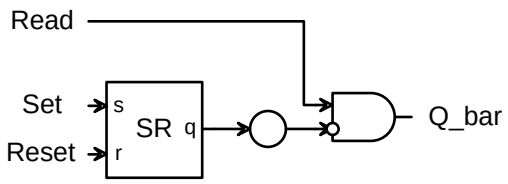


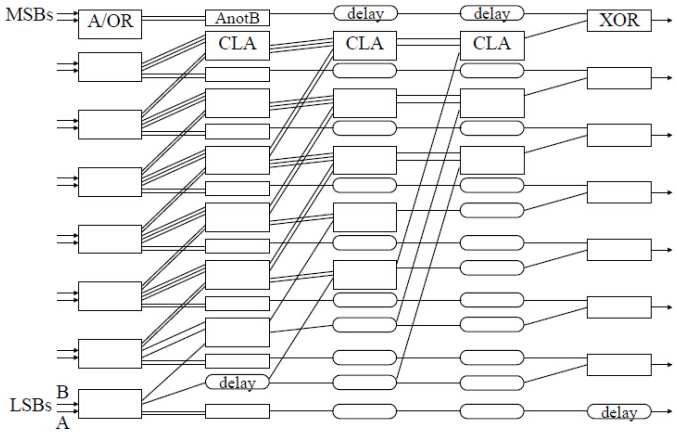
Figure 2.10: Non-destructive read-out storage cell schematic [27].

2.5 Fabricated RQL Design

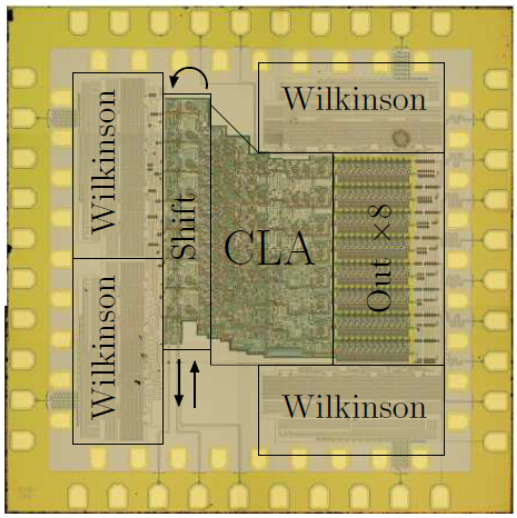
An RQL 8-bit carry look-ahead adder is designed by Northrop Grumman System Corp. in 2013 [28]. This is the first processing unit implemented and demonstrated with RQL technology.

This RQL carry look-ahead adder is implemented with the Kogge-Stone radix-2 structure. It contains $log_2 N + 2$ stages for an N-bits input as shown in Figure 2.11a on page 25. The first stage generates the prefix signals such as carry propagate and carry generate signals. The following module is the $log_2 N$ stages carry look-ahead network. In the final stage, the result is computed.

The latency of this adder is 150 ps (not including clock skew) at the clock frequency of 10 GHz, and the power dissipation is only 510 nW at 6.2 GHz. Figure 2.11b on page 25 shows the fabricated CLA chip on a 5×5 mm^2 die.



(a) Kogge-Stone CLA Structure.



(b) Kogge-Stone CLA microphotograph.

Figure 2.11: 8-bit Kogge-Stone CLA [28]. © 2013 IEEE.

2.6 SBU RQL VHDL Cell Library

In order to design and evaluate large-scale RQL circuit, simulation tools needs to be developed. A tunable RQL cell library was developed in the Ultra High Speed Computing (UHSC) Laboratory at Stony Brook University. This cell library logically models RQL circuits and provides functions for researchers to do cell level implementation, verification and statistic analysis for large-scale designs. Since the physical design tools and fabrication process is not available at this time, this library focuses on the layout-aware cell level design and provides design statistics that close to a completed physical design. Additional CAD tools are necessary to finish the physical design in the future.

All RQL cells are described using VHDL behavioral models with truth tables or finite state machines. Physical features such as energy consumption, JJs complexity, latency and approximate size are specified and tuned to the target fabrication process.

The cells have built-in functions to check any timing violations during simulation and report them. The timing violation report contains important information such as when and where the violation occurred, what kind of timing constraint is violated (the signals arrive too early or too late). This is helpful for a logic level designer to identify the source of the violation and figure out the best solution to resolve it.

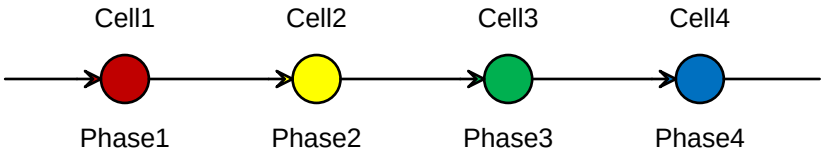
Switching activity of all cells is recorded during simulation and used to calculate dynamic power consumption. The library also provides functions for designers to obtain the design complexity in terms of JJs.

The design approach is successfully used in the joint work with HYPRES, Inc. on the development and demonstration of several 20 GHz RSFQ chips [48, 49, 51].

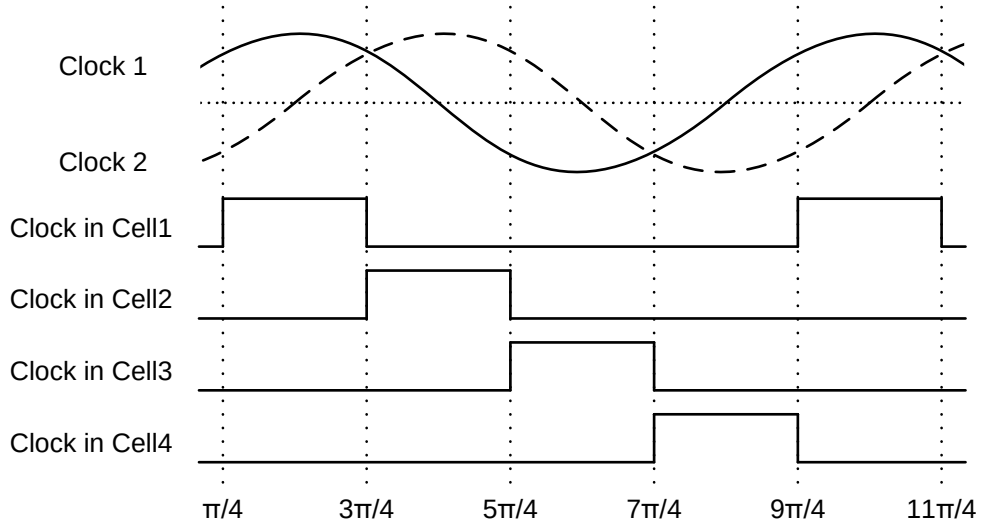
In this research, the cell library is tuned to the 248 nm 100 $\mu A/\mu m^2$ fabrication process from MIT Lincoln Lab. The features of this fabrication process will be provided in the next section.

2.6.1 Clock Model

As discussed in before, there are two clock & power lines in RQL, and cells that inductive coupled with clock & power lines can be set to one of the phases in a clock cycle. A cell can only process RQL pulses during the phase that it has been set. To describe this feature in the simulation, the library use standard logic '1' in VHDL to present the valid phase, and standard logic '0' to present other invalid phases. Figure 2.12a on page 27 shows an example circuit and the clock signals of each cell in the VHDL model. There are four cells in the example circuit, and are set to phase 1 (red), 2 (yellow), 3 (green), 4 (blue).



(a) Example circuit.



(b) Simulation waveforms of the clock signal.

Figure 2.12: Clock signal in VHDL model.

For Cell1 in phase 1, the clock signal is standard logic '1' from $\pi/4$ to $3\pi/4$, which means can process SFQ pulse during this time. For Cell2, Cell3, Cell4, the valid time to process signal are between$3\pi/4$ and $5\pi/4$, $5\pi/4$ and $7\pi/4$, $7\pi/4$ and $9\pi/4$, as shown in Figure 2.12b on page 27. This definition simplifies the behavioral of the clock and the timing violation checking.

2.6.2 Data Signal Model

As the data in RQL are represented by pulses (unlike voltage level in CMOS), a new definition is set up to represent data signals in SFQ pulse based circuits: the rising-edge is used to represent a positive pulse, and the falling-edge is used to represent a negative pulse, as shown in Figure 2.13 on page 28.

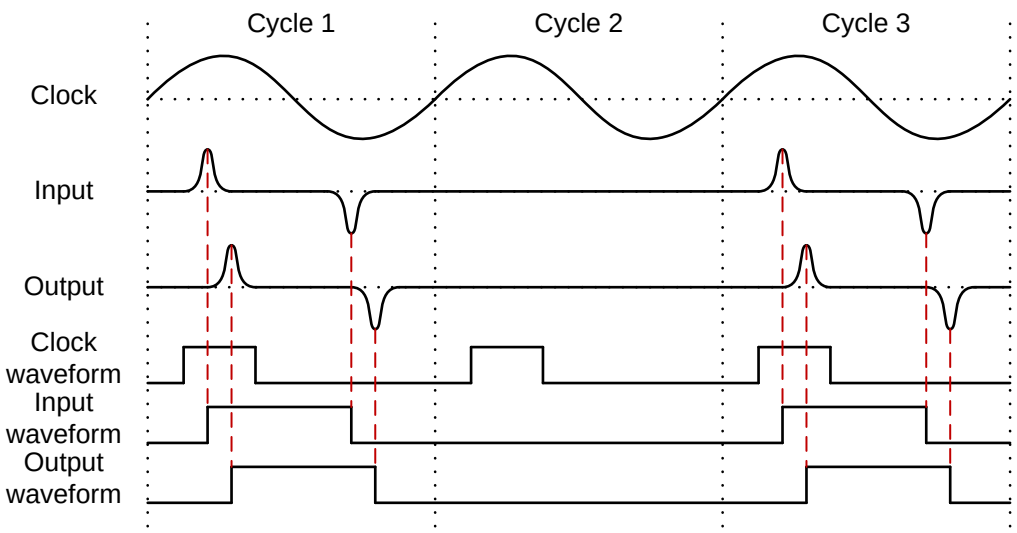


Figure 2.13: Data signal in VHDL model.

2.6.3 Acknowledgements

The SBU RQL VHDL cell library is a work finished by past and present members in the UHSC laboratory, Stony Brook University under the direction and supervision of Dr. Mikhail Dorojevets. The contributors are:

- Christopher Ayala (Major developer): Involved in all aspects of library development, including implementation, testing, documentation, and maintenance.
- Artur Kasperek: Implemented some gate-level FSMs and built-in functions.
- Kruti Shah and Prachi Bemalkhedkar: involved in the library documentation.
- Zuoting Chen (dissertation author): Implemented some additional features and built-in functions.

2.7 Target Fabrication Technology

The Massachusetts Institute of Technology Lincoln Laboratory has developed a fully planarized $100\mu A/\mu m^2$ process for very large scale integration (VLSI) SFQ circuit fabrication [52]. Figure 2.14 on page 29 shows several nodes of this fabrication process. This process is tune to support the development

Fabrication Process Attribute	✓	✓	Process Node		
	SFQ3ee	SFQ4ee	SFQ5ee	SFQ6ee	SFQ7ee
Critical Current Density (μA/μm²)	100	100	100	100	100
JJ diameter (surround) (nm)	700 (500)	700 (500)	700 (300)	500 (200)	500 (200)
Number of superconducting layers	4	8	10	10	10
Line width (space) (nm)	500 (1000)	500 (700)	350 (500)	250 (300)	180 (220)
Metal thickness (nm)	200	200	200	200	150
Dielectric thickness (nm)	200	200	200	200	180
Resistor width (space) (nm)	1000 (2000)	1000 (1000)	700 (700)	500 (500)	350 (350)
Resistor value (ohms per square)	2	2	2 and 0.002	2 and 0.002	2 and 0.002
Via diameter (surround) (nm)	700 (500)	700 (500)	500 (350)	350 (250)	350 (200)
Via type	Etched, Stacked Staggered	Etched, Stacked Staggered	Stud, Stacked	Stud, Stacked	Stud, Stacked
Process Development	Complete	Advanced	Underway	Underway	Underway
Early Access Availability	2013	Now	2015	2016	2017
Primary Process	Now	Sep. 2014	2016	2017	2018

Figure 2.14: MIT LL SFQ process [52]. Resent publication shows that SFQ5ee has 9 Nb superconducting layers [54].

of energy efficient circuit in the IARPA Cryogenic Computing Complexity (C3) program [53]. The process nodes, SFQ3ee and SFQ4ee with 4 and 8 superconducting layers, respectively, are already completed. Some advance process nodes (SFQ5ee, SFQ6ee, and SFQ7ee) with more superconducting layers and smaller JJ size are under development.

The SBU VHDL RQL cell library is tuned to the SFQ5ee process with 248 nm minimum feature size, 700 nm minimum JJ size, and 9 Nb superconducting layers (Figure 2.15 on page 30). The minimum JJ critical current is 38 μA. Compared to the earlier nodes, this process has smaller metal layer linewidth as well as etched via size. An additional high kinetic inductance superconducting layer is added below the first Nb layer M0 to enable compact bias inductors. A resistive layer is added to support interlayer sandwich-type resistors between Nb layers M4 and M5 [54].

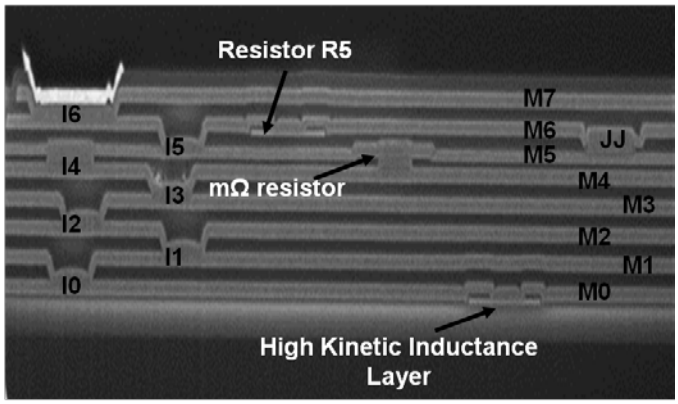


Figure 2.15: Cross section of the SFQ5ee process [54].

Chapter 3

Superconducting Memory and Research Goals

Outline

3.1 Brief Review of Superconducting Memory

The development of reliable, dense and fast energy efficient cryogenic memory remains to be one of the most serious challenges for superconductor computing. There have been several efforts to design such memory in the past, e.g., wholly SFQ memory in Japan [55, 56, 57], hybrid SFQ-CMOS in the USA [58] and JJ-MRAM, which is under development now [59, 60, 61]. However, these approaches are not suitable for the local storage units. Local storage units need to be designed in a new way to satisfy the requirements of frequency, latency and energy-efficiency.

This chapter first reviews the three memory approaches in superconducting memory design, then discusses the opportunities of designing local storage units with RQL. Finally, the research goal will be provided.

3.1.1 Wholly SFQ memory

In 1999, a Japanese research group developed a 16 Kbit superconducting latching/SFQ hybrid (SLASH) RAM with the frequency of up to 10 GHz. In this RAM, the decoders are composed of DC powered SFQ circuit, while the drivers and sense circuits have to be composed of AC powered latching devices. This RAM is demonstrated at low frequency, but fail at a high frequency of around 10 GHz because of the high-frequency AC powering [55, 56].

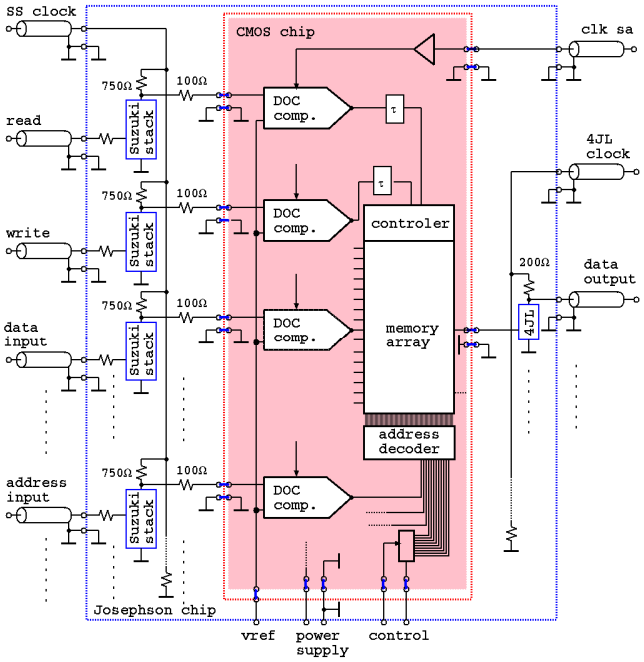
The next generation of the RAM is a RSFQ pipelined RAM. This is an all-DC-powered RAM with the capacity of 64 kbit - 1 Mbit. This RAM operates at 10 GHz and the maximum power dissipation is 12 mW for a 1 Mbit version.

3.1.2 Hybrid Josephson-CMOS Memory

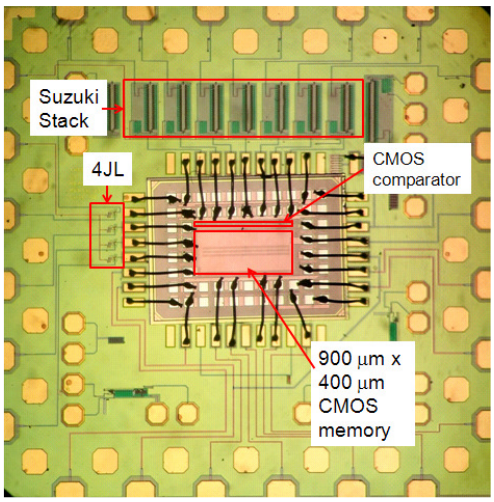
Hybrid memory is another approach to large capacity memory in superconductor technology by combining a room-temperature CMOS memory and a RSFQ interface circuit. A hybrid Josephson-CMOS memory was implemented in 2012 [58]. A 64 Kbit CMOS static random access memory (SRAM) is used as the storage component. In the hybrid interface circuit, a Suzuki stack (SS) with a four-junction logic (4JL) is used to amplify the millivolt superconductor logic signal and drives a sensitive CMOS comparator to produce volt level signals for CMOS circuit, as shown in Figure 3.1 on page 33.

The Josephson chip is fabricated using Hypres 4.5 kA/cm^2 niobium technology on a 5 mm $\times$ 5 mm die. The CMOS SRAM is fabricated using TSMC 65 nm technology on a 2.0 mm $\times$ 1.5 mm chip. These chips are connected by short wirebonds in a piggy-back package [58]. The access time is 400 ps and the power of read operation is 12 mW.

The latency and the energy efficiency of the hybrid memory are relatively high. And the major energy consumption comes from the interface circuit. As the result, hybrid memory is not a good approach to on-chip storage that requires low latency and high energy-efficiency.



(a) Hybrid RAM block diagram.



(b) Hybrid RAM microphotograph.

Figure 3.1: Hybrid Josephson-CMOS RAM [58]. © 2013 IEEE.

3.1.3 JJ-MRAM

Unfortunately, the two approaches discussed above have high energy consumption when costs of cryocooling are taken into account. To address this and

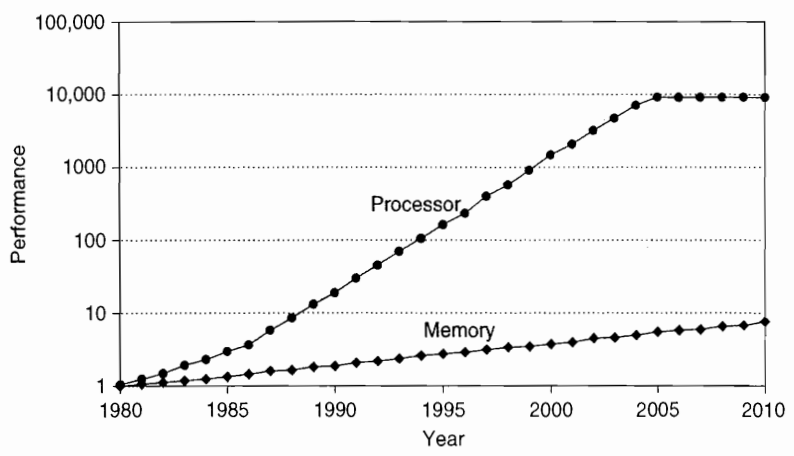


Figure 3.2: The performance gap between processor and memory [62].

other issues, a new type of cryogenic memory, energy-efficient high-density non-volatile JJ-MRAM, is under development [59, 60, 61]. This memory is built with magnetic Josephson junctions (MJJs). A MJJ is a JJ integrated with a ferromagnetic layer. Two distinctive states with high and low critical currents are provided in a MJJ, representing logical '0' and '1', respectively. This MJJ-based approach is electrically and physically compatible with SFQ circuits, allowing the fabrication of memories and processing units in the same chip.

3.2 New Opportunities in Energy-Efficient Local Storage Units Design with RQL

The local storage units are very important components in a microprocessor. The performance of the storage is always a bottleneck to processor performance. As shown in Figure 3.2 on page 34, the performance of the memory was 4 times less than the processor in 2010. To overcome this problem, designers not only make efforts to improve the storage performance in physical level, but also try to design new memory hierarchy to improve the efficiency of the data transmission. Most of the modern processors contain local storage units to reduce the latency of the datapath and improve the performance. Most of the CMOS local storage units are built with standard cell flip-flop or SRAM to achieve high frequency and low latency.

On-chip storage (e.g., register files) has a different set of requirements than main memory. In RQL processors, on-chip storage units need to operate at the same clock frequency as processing units, have low latency, and provide

high data bandwidth through a use of multiple read and write 32-/64-bit wide ports. Compared to main memory, the on-chip storage size is much smaller, sometimes only few Kbits per unit. All of this dictated a use of the same circuit technology, namely RQL, for on-chip storage implementation.

Another advantage of using RQL in local storage units design is that when the fabrication technology improves, the characteristics of RQL local storage units can be improved in the same way as other processing units in a RQL processor.

3.3 Research Goals

The major objective of this work is to study the use of RQL for high-performance energy-efficient on-chip storage design to get important insights into practical RQL memory design for future superconductor processors. We designed and analyzed four types (13 in total) of 32- and 64-bit RQL multi-ported pipelined local storage units that are placed alongside processing units, namely: 1) random access memories (RAM), 2) register files, 3) directed-mapped write-through and write-back caches, and 4) first-in first-out buffers. All storage units in this research are designed to operate at the frequency of 8.5 GHz and optimized to achieve low latency and low energy consumption. Three key characters are analyzed: latency, design complexity and energy consumption. Latency is the time required to generate output data or signal during an operation, which represents the performance of a design. Design complexity is defined as the number of Josephson junction required in a storage design. Energy consumption profile reflects the contribution of all sub-blocks in each unit for every type of operations. Scaling of these designs is also evaluated. Since the target fabrication process is still under development, our evaluation only focus on the variation of storage capacity. The energy costs of cryo-cooling are not included.

Chapter 4

Local RAM with 1 Read and 1 Write Ports

Outline

4.1 Design Overview

The first memory designed in this research is RQL RAM with 1 read and 1 write ports. Since RQL storage requires different the design rules than those used for CMOS memory, new techniques should be explored. In this study, our goal is to figure out the low-latency and energy-efficient solutions to design 32- and 64-bit RQL RAM with the capacity of 1 to 4 Kbit. Key characteristics, such as latency, design complexity and energy consumption, are collected during the simulation. The results from this design provide the

first view on RQL storage and the techniques employed in this design can help us in more complex designs later on.

There are three types of local RAM been implemented: a 1 Kbit RAM (32-word × 32-bit), a 2 Kbit RAM (64-word × 32-bit), and a 4 Kbit RAM (64-word × 64-bit). These RAM are targeted at the frequency of 8.5 GHz.

4.2 RQL RAM Design

The dual-ported RQL RAM shown in Figure 4.1 on page 38 can perform one read and one write operation per cycle. It has two major parts: two address decoders (one for read and one for write operations) and multiple data slices.

An address decoder is implemented with a use of predecoding [63] (which will be discussed later). The decoder has two parts: predecoder, which is on the top of a RAM; and final stage decoder, which is in the middle of a data slice. A RAM data slice consists of 1) a storage cell array, 2) separate read and write wordlines, and 3) separate read and write bitlines. (This is different from CMOS memory where same wordlines and bitlines are used during read and write operations.) The storage cells in a slice are grouped into several data arrays, each of them has the size of 8-word × 4-bit. Table 4.1 on page 37 shows major components of three types of RAM. Since there is only one word in a row, depth is defined as number of words in a RAM.

Table 4.1: Major RAM design components.

Capacity	1 Kbit	2 Kbit	4 Kbit
Data width, bits	32	32	64
Depth, words	32	64	64
# of read decoders	1	1	1
# of write decoders	1	1	1
# of data slices	4	8	8
# of data arrays per slice	8	8	16

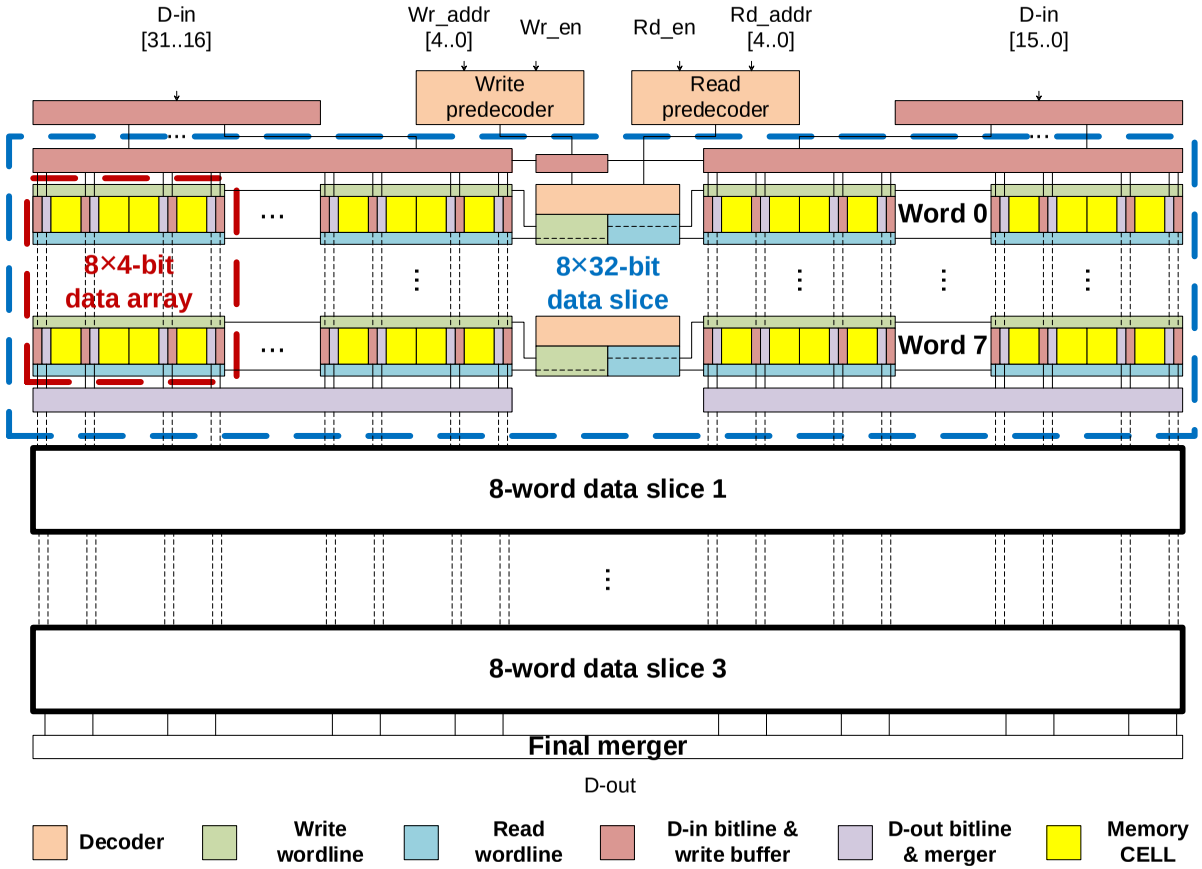


Figure 4.1: Top-level structure of a 1 Kbit memory.

4.2.1 Decoder

There are two decoders implemented in the design, one for read and another for write operations. In order to reduce the decoder area, the technique of predecoding [63] is employed. Each decoder has two parts: predecoders placed on the top of a RAM, and several final stage decoders located in the middle of data slices, one per data slice. An address is broken into two fields: slice address and word address within a slice. These two addresses are predecoded simultaneously by their predecoders into a one-hot slice and word indices. These indices serve as inputs to final stage decoders. The final stage decoders generate word select signals that are sent through wordlines to select one row of storage cells.

In 1 Kbit RAM, the higher 2 bits of the read / write address are predecoded into 4-bit one-hot read / write slice index to select one of the data slices. The lower 3 bits are predecoded into 8-bit one-hot read / write word index to select one word in a data slice. The slice index and the word index are ANDed in the final stage decoder, as shown in Figure 4.2 on page 39. For 2 Kbit and 4 Kbit RAM, the read / write address has 6 bits: higher 3 bits for slice address and lower 3 bits for word address.

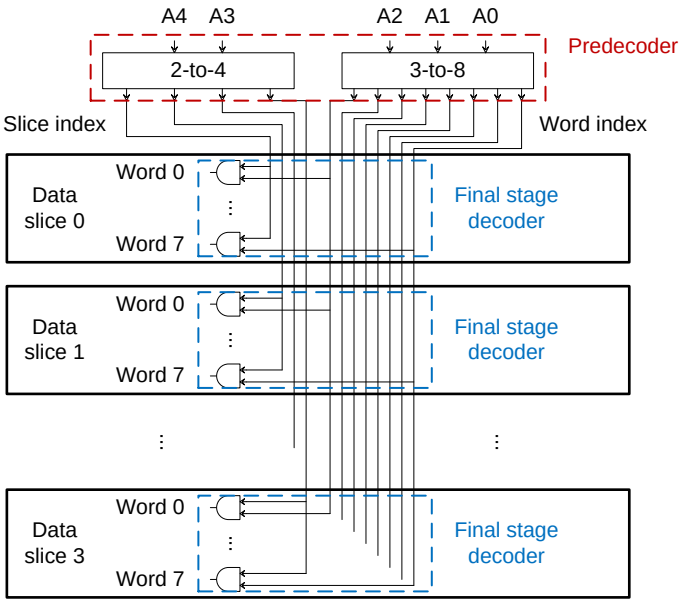


Figure 4.2: A 1 Kbit RAM decoder schematic.

4.2.2 Data Slice

A 32-bit data slice has a final stage decoder in the middle, 8 8-word $\times$ 4-bit data arrays (4 on each side), a write buffer on the top and a data merger on the bottom. For 64-bit data block, there are 16 data arrays in total (8 in each side).

The first problem we face is how to broadcast a word select signal to 32/64 storage cells in every row, same as any other lines with fan-out higher than 1. Unlike CMOS, RQL is a pulse-based logic and the only cell that can serve as a pulse splitter is a connection cell with fan-out of 2. To overcome this problem, a binary tree-like broadcasting structures is employed to achieve the require fan-out and have a low broadcasting latency. For a long distance propagation crossing multiple data arrays, a connection cell-PLT-PTL receiver structure is used, taking advantage from the fast and energy lossless features of a PTL.

The wordline broadcasting circuit has a two-level hierarchy: global and local. To broadcast signal to a 32-bit row, for example, the global wordline first splits the signal into 8 using a connection cell-based binary tree, then sends these signals to 8 local wordlines via PTL. The local wordline in the data array receives the signal from the global wordline and splits it into 4, then sends to each storage cell.

To deliver input data to cells in storage array columns, vertical D-in bitlines are implemented with broadcast tree-like structure similar to the one used for the wordlines. D-in bitlines also have a two-level hierarchy. Input data are propagated to each data slice through global D-in bitlines and reach each storage cell in a column via local D-in bitlines. To reduce energy consumption, a write buffer is placed on the top of each data slice to prevent data from going through local D-in bitlines except for the slice that is selected by a slice index.

The second problems is, in RQL, there are no tri-state buffers (like the ones used in CMOS memories) to connect storage cells to (vertical) D-out bitlines to collect and propagate output data. These bitlines have to be implemented with RQL OR gate chains to merge output data vertically along storage cell columns. To reduce energy consumption and latency, D-out bitlines also have a two-level hierarchy: global (inter-slice) and local (intra-slice). Output data are collected by local D-out bitlines, and propagated to the output via global D-out bitlines.

The schematic of a data slice and a data array are shown in Figure 4.3 on page 41 and Figure 4.4 on page 41, respectively.

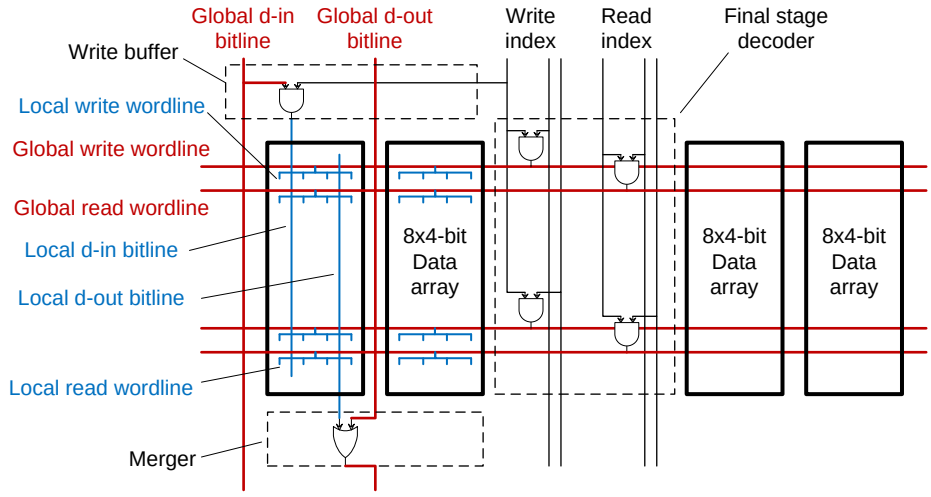


Figure 4.3: Memory data slice.

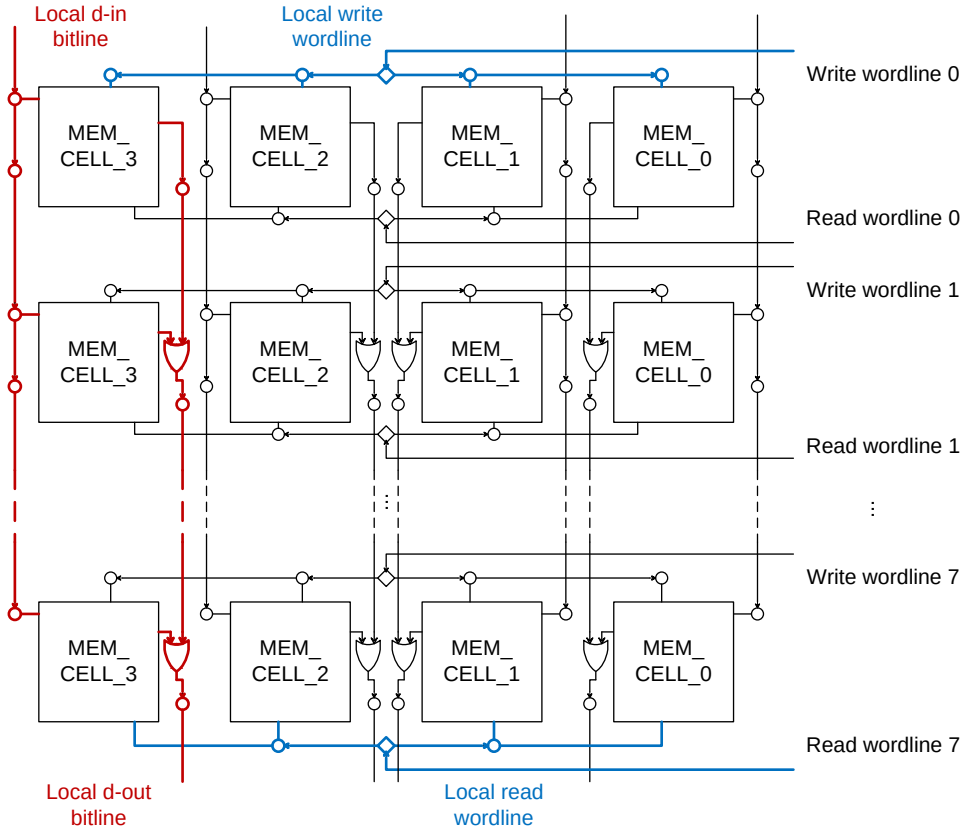


Figure 4.4: Memory data array.

4.2.3 Memory Macrocell

RQL Non-Destructive Read-Out (NDRO) cells were used to build a memory macrocell in RAM. A 1-bit 23-JJ memory macrocell contains a NDRO cell and additional control and interconnect logic, as shown in Figure 4.5 on page 42. We found that the most efficient way (in terms of size, complexity and energy consumption) to implement write operations in NDRO-based storage was to use a preset-to-zero approach. When executing a write operation, a write enable signal sets the NDRO cell to '1' during the first half of the cycle. If the input data is logical '1', the macrocell control logic generates a signal to reset the cell to '0'. A read signal reads a logical '0' when the internal state of the cell is '1', and logical '1' when the internal state is '0'.

A read access is finished within one phase (cross the boundary of the first and second phase), and a write access is finished in three phases. As the result, if both read and write operation happen at the same clock cycle, the memory macrocell would output the old data, and then store the new data. Input data is always available for the read access in the next cycle.

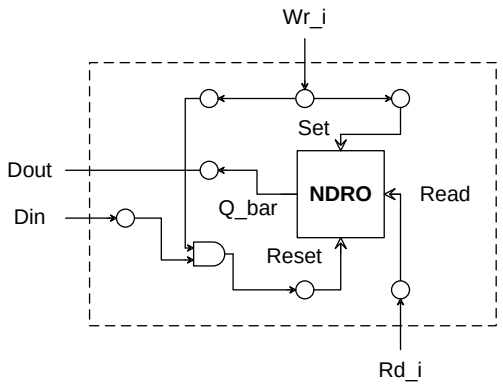
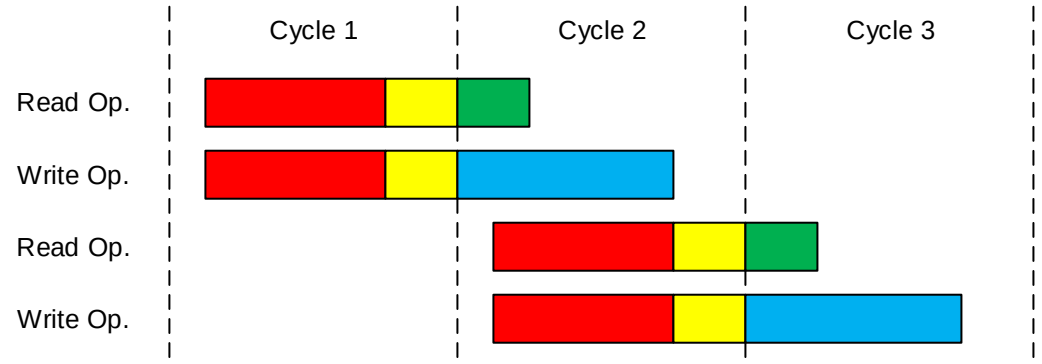


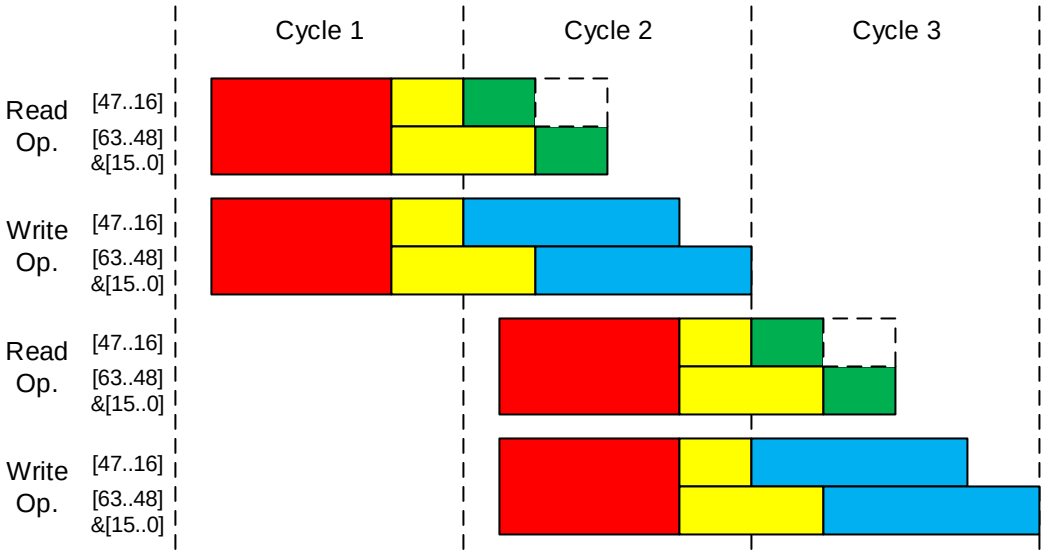
Figure 4.5: Memory macrocell.

4.2.4 Pipeline Structure

In order to increase the throughput, pipelining is employed in these designs. The pipeline of the memory is shown in Figure 4.6 on page 43. For read operation, there are 3 stages: decode, wordline broadcast, and NDRO read & d-out. For write operation, there are also 3 stages: decode, wordline broadcast and NDRO write. Both read and write operations start in the middle of the first quarter cycle (phase 1), because all the signals are transmitted by PTL, and the reception window of the receiver is in the middle of the phase, as



(a) RAM pipelines (32-bit data width).



(b) RAM pipelines (64-bit data width).

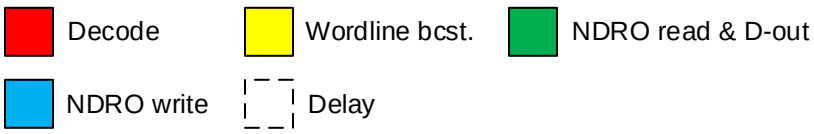


Figure 4.6: RAM pipelines (not incl. clock skew).

described in 2.4.1. The NDRO gate takes less than a quarter cycle (one phases) to finish the read operation but three quarter cycles (three phases) to finish the write operation.

For a 64-bit version, time to propagate the wordline signals to the least significant bit (LSB) and the most significant bit (MSB) is two phases. In order to reduce the latency, the NDRO cells access of one word is broken into two parts: bit 47 to bit 16, which has the same timing as a 32-bit RAM, and bit 63 to 48 & bit 15 to 0, which has one additional quarter cycle in wordline broadcast stage.

If both read and write operations access the same address in the same cycle, the old data inside the memory would be read, and the new data will be stored in the memory. This new input data are available for the next read operation.

4.2.5 Critical Path

The critical signal propagation path in the memory is shown in Figure 4.7 on page 44. For the read operation, the critical path is from the read decoder to the top wordline, and the d-out bitline from the least or most significant bit. For a write operation, the critical path is from the decoder to the LSB or MSB in the bottom word.

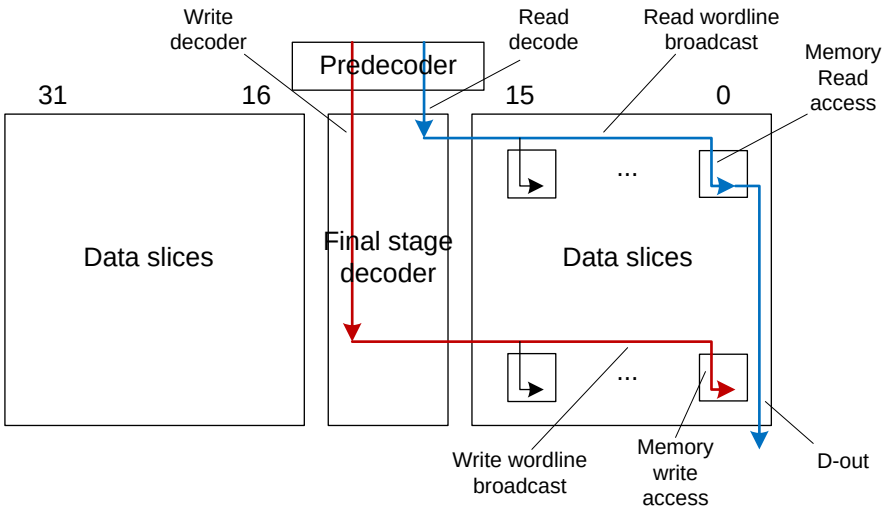


Figure 4.7: Critical path in a RAM.

Table 4.2: Summary of the RAM designs.

Data capacity, Kbits	1	2	4
Data width, bits	32	32	64
Depth, words	32	64	64
Clock frequency, GHz	8.5	8.5	8.5
Complexity, JJs	42512	84132	167876
Read latency (incl. clock skew), ps	204.50	249.30	278.80
Write latency (incl. clock skew), ps	235.50	277.12	306.62
Average energy/op, aJ	111.21	141.06	269.68

4.3 Simulation Results and Discussion

The simulation is focus on study the scaling of the designs as the capacity increase. There are three key characters been evaluated: latency, design complexity and energy consumption. Area is not included in the study because it is highly depend on the specific fabrication process, which is not available at this time. The simulation summary is shown in Table 4.2 on page 45.

All memories are verified using random test vectors generated by the testbench. All the statistics are collected with the minimum critical current of 38 µA at the temperature of 4.2 K.

4.3.1 Latency

As the capacity grows from 1 Kbit to 2 Kbit, the depth of the RAM is doubled. Both read and write latencies are increased by ~40 ps because bitlines in 2 Kbit is longer than that in 1 Kbit. When increasing memory capacity from 2 to 4 Kbit, the depth remains unchanged but the width is doubled. Both read and write latencies are increased by approximately one quarter clock cycle (one phase). This is because the time to propagate the signal over wordlines is a half cycle (two phases) in the 4 Kbit RAM compared to one quarter cycle (one phase) in the 2 Kbit RAM.

4.3.2 Design Complexity

Figure 4.8 on page 48 shows the complexity of three types of memories. Memory marcocells make the major contribution, taking more than 50% of all the junctions in the designs. As the depth doubles (from 1 Kbit to 2Kbit), all

components are approximately doubles. As the width increase from 32-bit (2 Kbit RAM) to 64-bit (4 Kbit RAM), all components except decoders are doubled. This is because complexity of decoders only depends on the depth of the RAM.

Notice number of junctions required for read wordlines is ~2/3 of junctions required for write wordlines. This is due to the optimized interconnect of memory macrocell: read enable inputs (Rd_i in Figure 4.5 on page 42) of memory macrocells are closer than write enable inputs (Wr_i in Figure 4.5 on page 42), which requires less connection cells to propagate signals.

4.3.3 Energy Consumption

To study energy consumption, four test cases are applied, namely: 1) read only, which performs one read operation per cycle; 2) write only, which performs one write operation per cycle; 3) read + write, which performs one read and one write operation simultaneously per cycle; and 4) no op, which does not perform any read or write operation during the test. In these cases, RAM are initially filled with random data. During the first three tests, all address (read address and write address) and input data (if any) are random vectors generated by testbench. Figure 4.9 on page 49, Figure 4.10 on page 50 and Figure 4.11 on page 51 shows the energy consumption in three test cases: read only, write only and read + write, respectively. Figure 4.12 on page 52 shows the percentage of dynamic and stand-by energy in total energy consumption in terms of each RAM designs and test case.

By comparing the energy of read operation (Figure 4.9 on page 49) and write operation (Figure 4.10 on page 50), we can see that write operation consumes ~2.3x more energy than read operation. This difference mostly comes from memory macrocells and bitlines (d-in bitlines & write buffer and d-out bitlines & mergers). In a memory macrocell, a complex control circuit is used to transform write enable and input data to set and reset signals to switch Set/Reset gate during a write access. In contrast, output data can be naturally read by a read enable signal. The energy difference in bitlines results from a binary-tree-like broadcasting circuit in write buffers. This broadcasting circuit is used to broadcast block select signal to all bits in a data block and enable input data to reach local d-in bitlines, as shown in Figure 4.3 on page 41.

As the depth doubles (capacity increase from 1 Kbit to 2 Kbit), the energy cost for d-in bitlines & write buffers is increase by ~30% and d-out bitlines and mergers is increase by ~20%. The reason for this is number of data blocks in 2 Kbit RAM is double compared to 1Kbit RAM, and data propagation distance in bitlines is increased. As the width doubles (capacity increase from 2 Kbit

to 4 Kbit), the energy cost on memory macrocells, d-in bitlines & write buffer and d-out bitlines & mergers are doubled as expected.

In Figure 4.12 on page 52, dynamic energy is defined by the energy spent when a JJ switches in read decoder, write decoder, memory macrocells, d-in bitlines & write buffers, d-out bitlines & mergers, read wordlines and write wordlines. Stand-by energy is defined as the energy spend on inactive (no switch) junctions. We can see that dynamic energy make more contribution on total energy consumption in write operation compared to read operation. This means more junctions are switched during a write operation than a read operation. For read + write case, the number of switched junction is even more. As the capacity grows, the percentage of stand-by energy is increase, because more stand-by energy is required to store the data in a RAM.

To further evaluate the stand-by energy, we introduce another test case: "no op". In this test case, all cells are inactive and only stand-by energy is consumed. The result is shown in Table 4.3 on page 53. "% of stand-by JJs" is given by stand-by energy of three test cases (read only, write only and read +write) divide by stand-by energy of no op case. We can see from the table that only small amounts of JJs are switched during an operation.

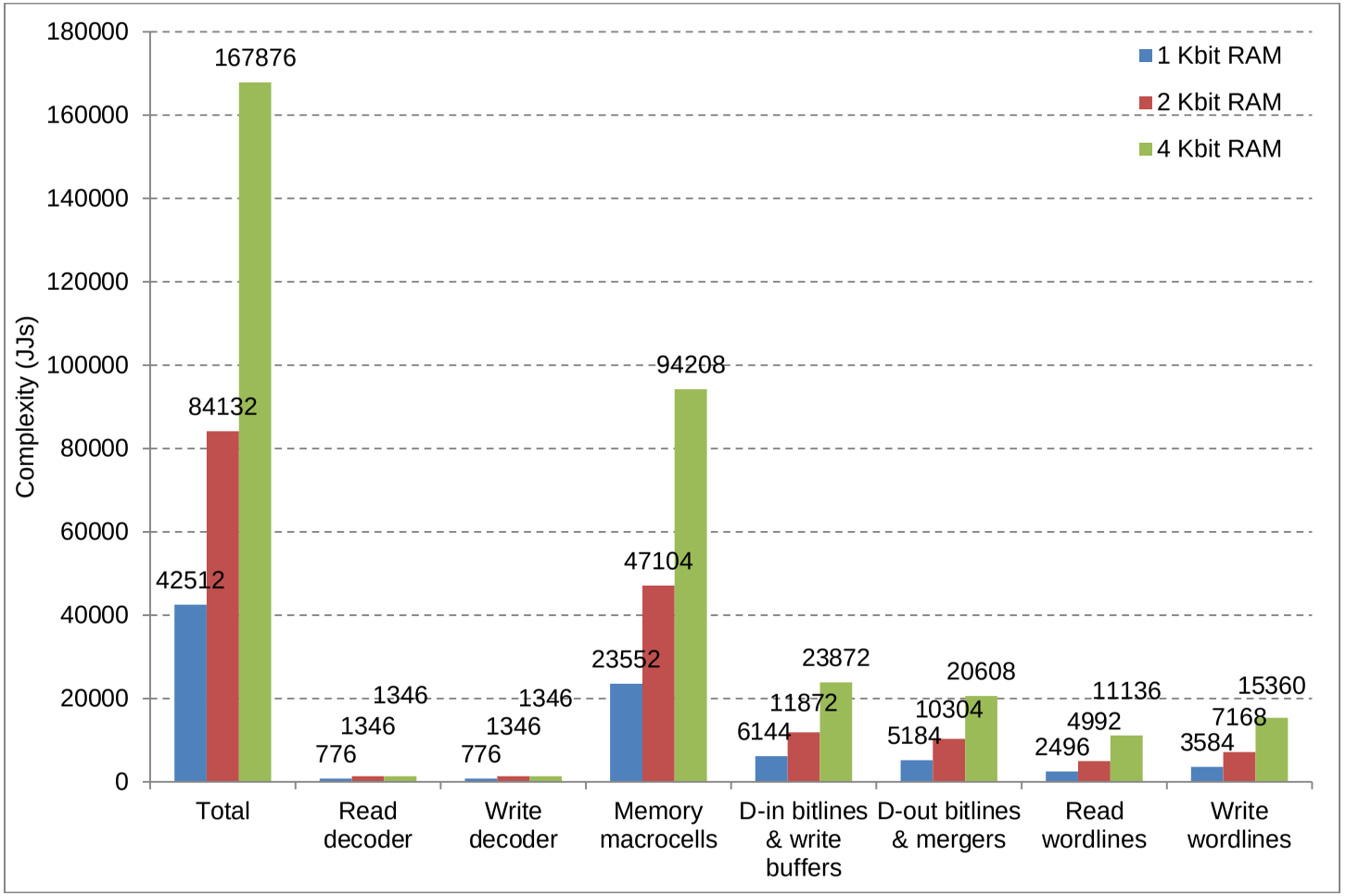


Figure 4.8: RAM complexity breakdown.

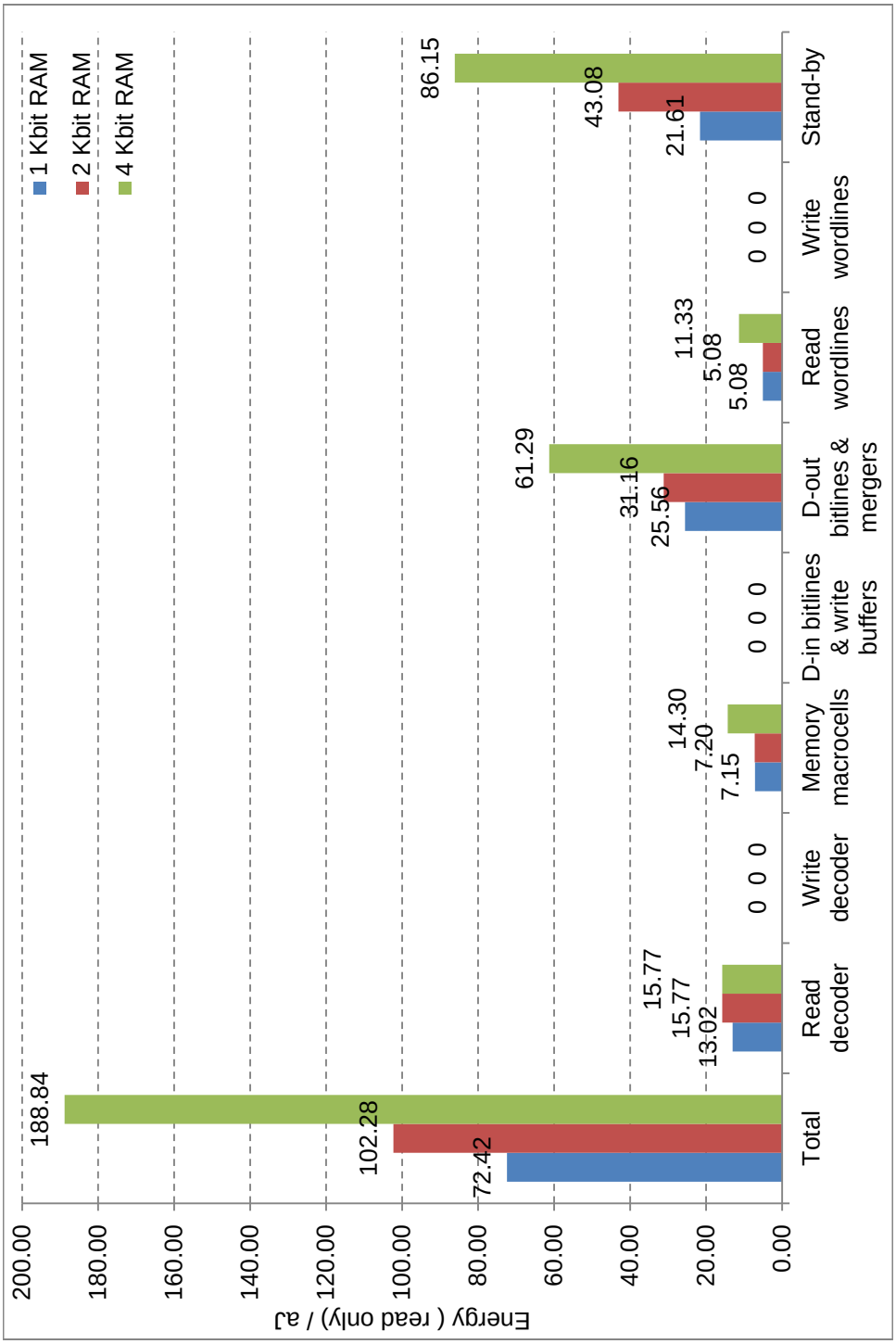


Figure 4.9: RAM energy per read only operation.

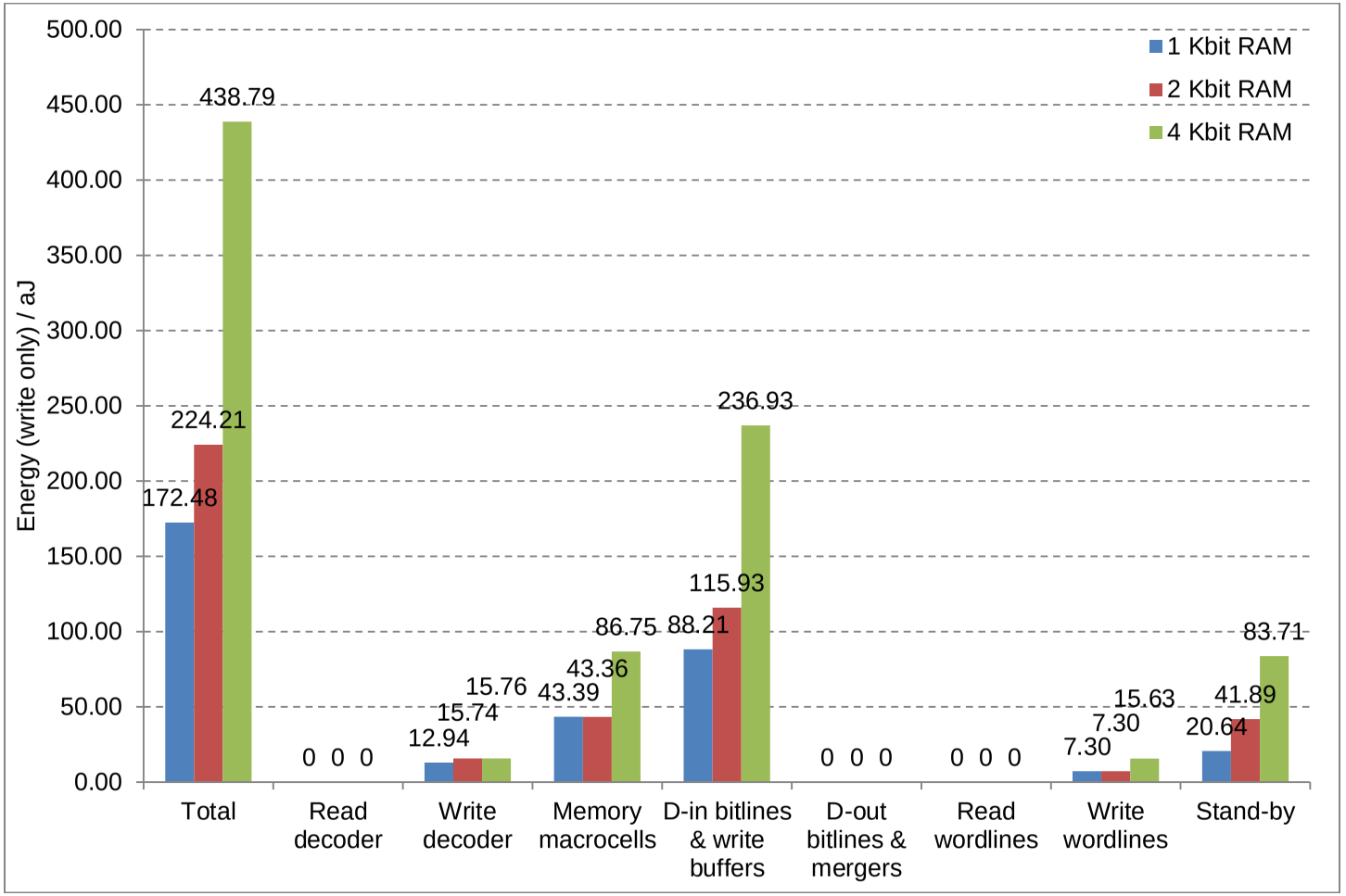


Figure 4.10: RAM energy per write only operation.

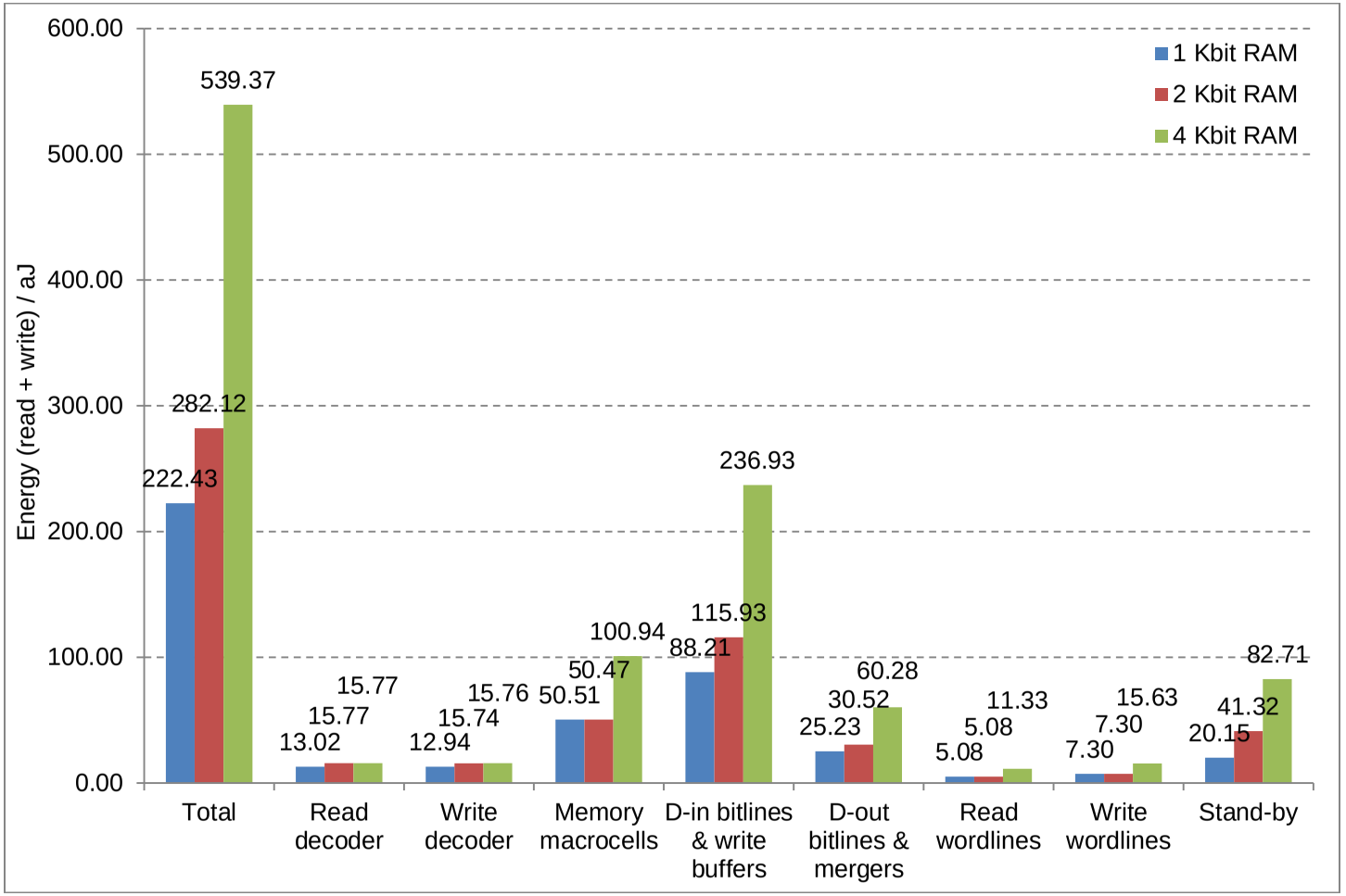


Figure 4.11: RAM energy per read + write operation.

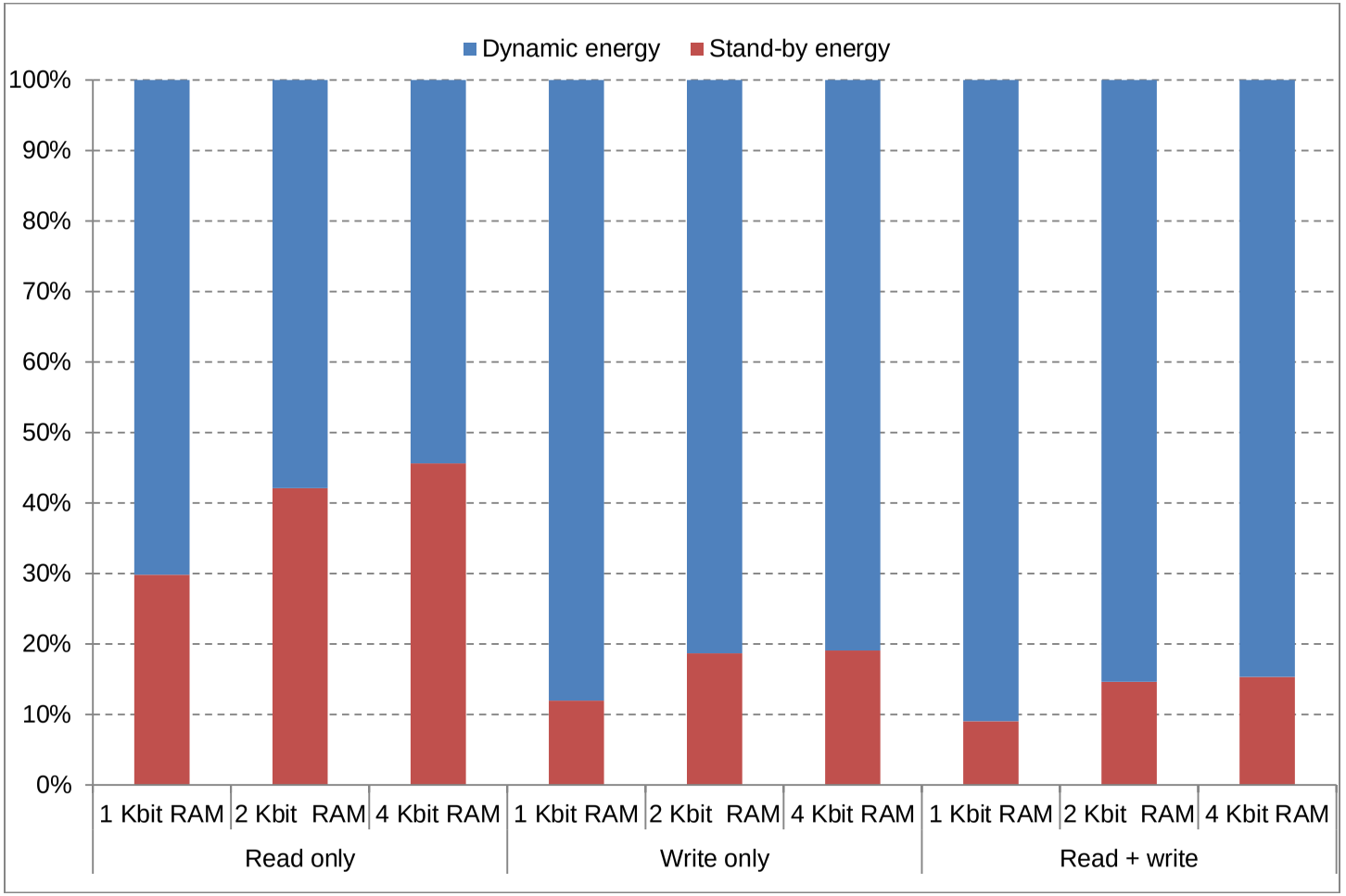


Figure 4.12: RAM dynamic and stand-by energy breakdown.

Table 4.3: RAM stand-by energy.

Test case	**1 Kbit RAM**		**2 Kbit RAM**		**4 Kbit RAM**	
	Stand-by energy (aJ)	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**
Read only	21.61	97.76%	43.08	98.67%	86.15	98.85%
Write only	20.64	93.38%	41.89	95.95%	83.71	96.05%
Read + write	20.15	91.15%	41.32	94.64%	82.71	94.91%
No op	22.10	100.00%	43.66	100.00%	87.15	100.00%

Chapter 5

Register Files with 2 Read and 1 Write Ports

Outline

5.1 Design Overview

Register files with 2 read and 1write ports are designed in this chapter. A register file has basically the same organization as RAM, with an additional read port and corresponding wordlines and bitlines added to every data slice. This allows the register file to execute two read and one write operation per cycle.

In order to support the second read operation, we designed a new type of non-destructive read-out storage cell, namely, NDRO2. With 2 read and 1 write ports, the NDRO2 cell can be naturally used in the register file without any additional control logic.

Table 5.1: Major register file design components.

Capacity	1 Kbit	2 Kbit	4 Kbit
Data width, bits	32	64	64
Depth, words	32	32	64
# of read decoders	2	2	2
# of write decoders	1	1	1
# of data slices	4	4	8
# of data arrays per slice	8	16	16

Same as RAM, key characters such as latency, design complexity and energy consumption are studied as the capacity scales. The result from this study allows us to analyze the trade-off of adding the second read port.

There are three types of register files been implemented: a 1 Kbit register file (32-word $\times$ 32-bit), a 2 Kbit register file (32-word $\times$ 64-bit), and a 4 Kbit register file (64-word $\times$ 64-bit). All the register files are targeted at the frequency of 8 GHz.

5.2 RQL Register File Design

With the same structure as RAM, register file also has two major parts, three decoders and several data slices. Table 5.1 on page 55 shows the major components in register files. These register files have two read decoders compared to RAM. The structure of the 1 Kbit register file is shown in Figure 5.1 on page 56.

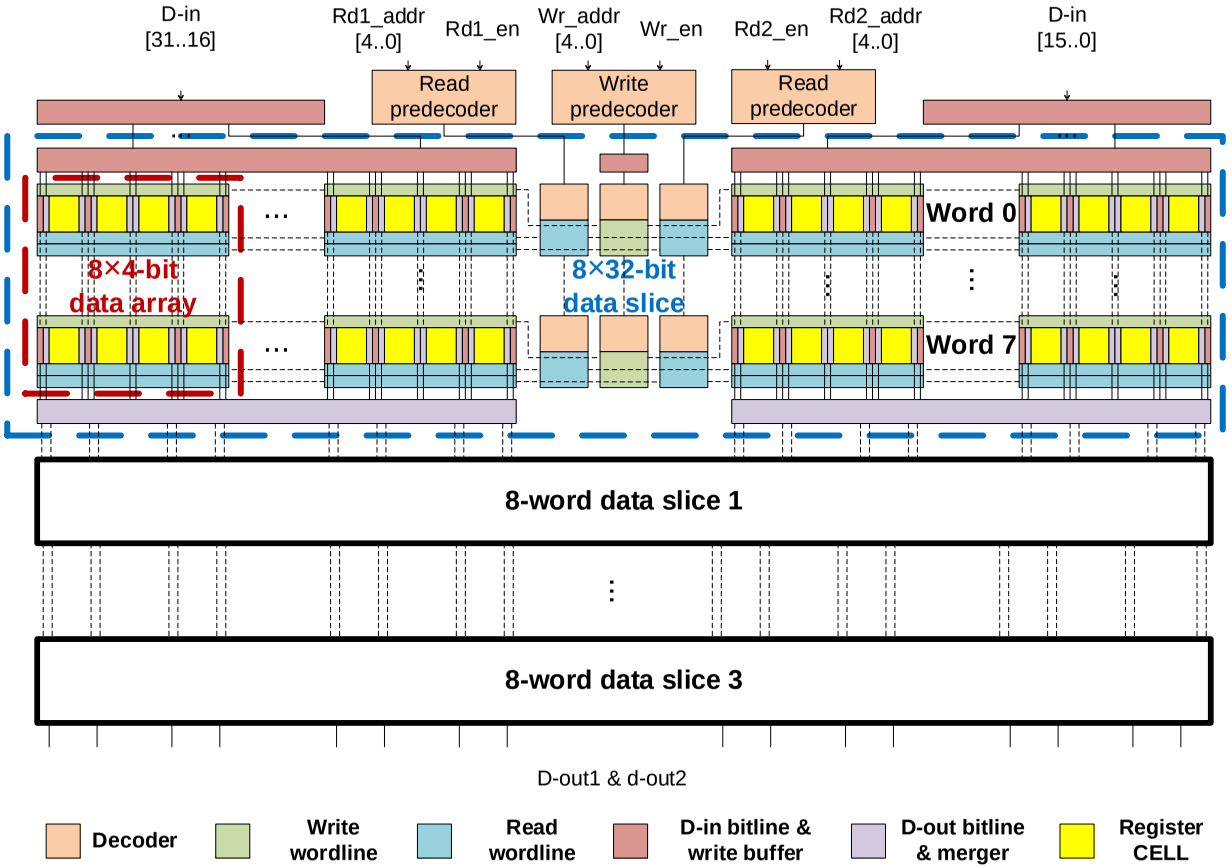


Figure 5.1: Top-level structure of a 1 Kbit register file.

5.2.1 Data Slice

The schematic of the register data slice is shown in Figure 5.2 on page 57. The final stage of the second read decoder is placed in the middle of the data slice. The new sets of wordlines and bitlines employ the same broadcasting and merging strategies as the wordlines and bitlines in the RAM. Figure 5.3 on page 58 shows the local wordlines and bitlines in the register data array.

With the new decoder and the wordlines and bitlines, the width and height of the data slice is increase compared to RAM. Despite that, we can still manage to finish broadcasting in one quarter cycle for the 32-bit version and two quarter cycles in the 64-bit version.

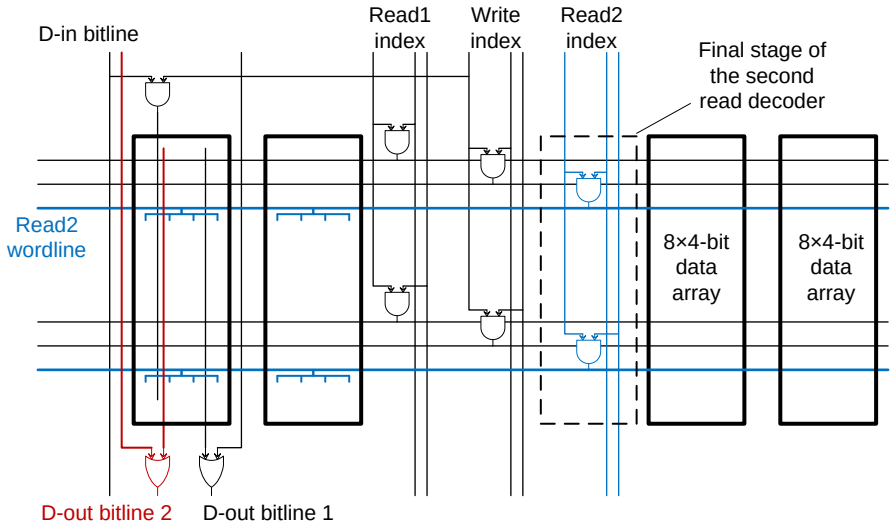


Figure 5.2: Register file data block.

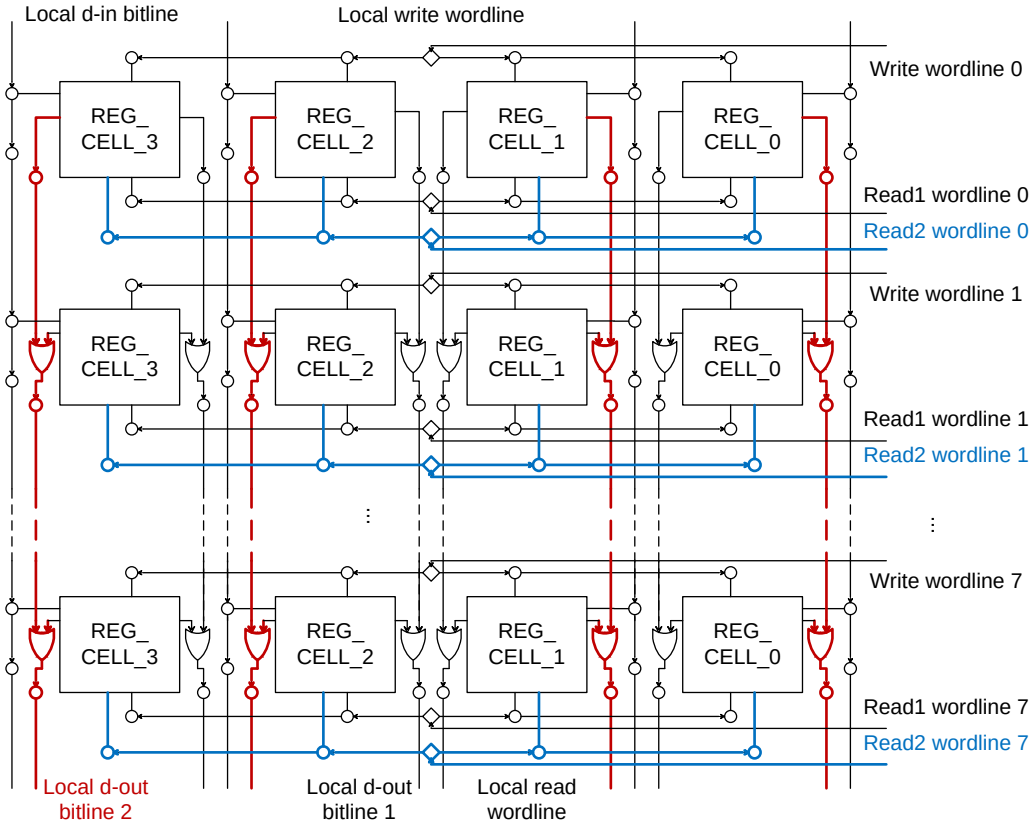


Figure 5.3: Register file data array.

5.2.2 Register Macrocell

The first approach to the register macrocell is based on the 1-read 1-wrtie port NDRO cell. An additional demultiplexer is used to drive the output from a NDRO cell into two different read ports. Additional control logic is necessary for demultiplexer, which dramatically increases the latency, design complexity and energy consumption of the register file. To solve this problem, we found out that the best approach was to modify the NDRO storage cell to add another read port. A new 13-JJ NDRO2 cell with two read and one write ports was used in the implementation of a 1-bit register macrocell with 33 JJs, as shown in Figure 5.4 on page 59.

A register macrocell schematic is shown in Figure 5.5 on page 59. With 2 read and 1 write ports, NDRO2 cell can be easily used in the register file without any additional control logic. A register macrocell has the same timing behavior as a memory macrocell.

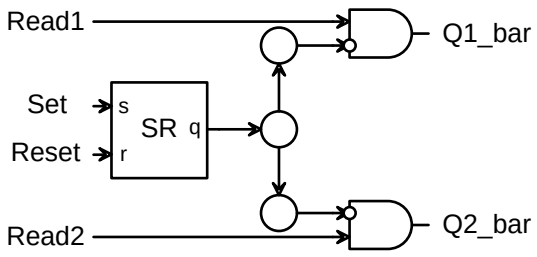


Figure 5.4: NDRO2 schematic.

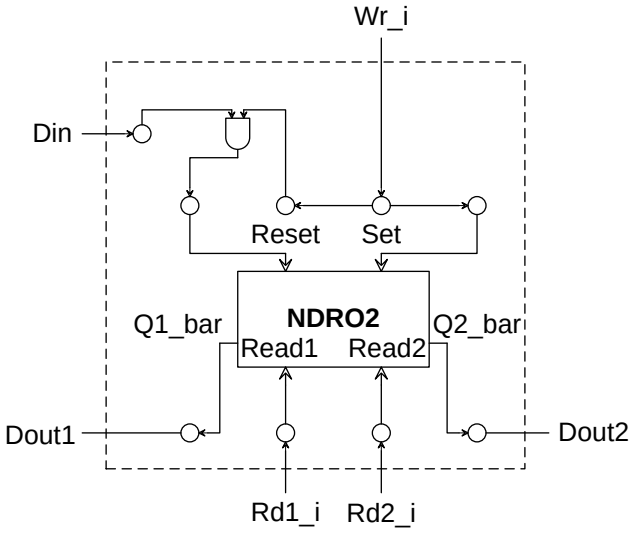


Figure 5.5: Register macrocell.

5.3 Simulation Results and Discussion

Similar to RAM in Chapter 4, this simulation focuses on evaluating how capacitance influences latency, design complexity and energy consumption. In addition, the effect of adding a new read port is analyzed. The simulation summary is shown in Table 5.2 on page 60.

Functionality of register files is verified using random test vectors generated by the testbench. All the statistics are collected with the minimum critical current of 38 µA at the temperature of 4.2 K.

5.3.1 Latency

As the width grows from 32 to 64 bits (from 1 Kbit to 2 Kbit), both read and write latency are increase by one quarter clock cycle (one phase), which is the same as RAM. As the depth doubles, the read latency is increased by ~70 ps and write latency is increase by ~65 ps. This is result from longer propagation distance in bitlines.

Comparing with RAM, both read and write latency are increased. For example, 1 Kbit RAM has a read latency of 204.50 ps and a write latency of 235.50 ps. In 1 Kbit register file, the read latency is increased to 226.53 ps and the write latency is increased to 260.74 ps. This is because the additional read port increases the size in both width and height, which increase the propagation delay of signals.

Table 5.2: Summary of the register file designs.

Data capacity, Kbits	1	2	4
Data width, bits	32	64	64
Depth, words	32	32	64
Clock frequency, GHz	8.5	8.5	8.5
Complexity, JJs	64578	127426	253918
Read latency (incl. clock skew), ps	226.53	256.03	327.19
Write latency (incl. clock skew), ps	260.74	290.24	355.70
Average energy/op, aJ	102.38	191.86	246.77

5.3.2 Design Complexity

Figure 5.6 on page 63 shows the complexity of three register files. As the capacity doubles (depth is doubled or width is doubled), all components except decoders are doubled. The complexity of the decoders only depends on the depth of the register file.

Figure 5.7 on page 64 shows the normalized complexity of a 1 Kbit RAM against a 1 Kbit register file. The complexity of read decoders and d-out bitlines & mergers are approximately doubled as expected. These come from the second read port in register file. The complexity of the storage macrocells (memory macrocells in RAM and register macrocells in register file) is increased by 1.43 times. This is the cost of using NDRO2 based cell in register file. The complexity increase of read wordlines is expected to be 2x since read wordline is doubled. However, the complexity of read wordline is increased by ~3x. The reason of this is the size (both width and height) of a register macrocell is larger than a memory macrocell and the distances between read enable inputs (rd1_i, rd2_i in Figure 5.5 on page 59) are increased. Read wordlines, which are optimized for RAM (discussed in Chapter 4), are not suitable for register macrocell array. More connection cells are required to transmit RQL pulse as the distance grows. The complexity of write wordlines do not influence by the growth grows because write wordline broadcasting circuits in RAM are capable to use in the register file without timing violation.

5.3.3 Energy Consumption

Four test cases are applied to study energy consumption: read only (two read operations per cycle), write only (one write operation per cycle), read + write (two reads and one write operations per cycle) and no op (no read or write operation during the test). Register files are initially filled with random vectors. All input vectors during the test are random. Figure 5.8 on page 65, Figure 5.9 on page 66 and Figure 5.10 on page 67 shows the energy consumption in read only, write only and read + write cases, respectively.

Results in those figures show the similar conclusions as RAM: a write operation consumes more energy than a read operation. As the width doubled, the energy of register macrocells, d-in bitlines & write buffers, d-out bitlines & mergers, read wordlines and write wordlines are doubled. As the depth doubled, the energy of d-in bitlines & write buffers, d-out bitlines & mergers, read wordlines and write wordlines are increased.

Figure 5.11 on page 68 shows the normalized energy of a 1 Kbit RAM against a 1 Kbit register file in read + write case. Total energy is increased by 1.38x with the added read port. For read decoders, the energy is double as

expected since there are two read decoders in a register file. The energy spent on the storage macrocells are 1.31x higher in register file than RAM. This is the cost of using NDRO2 based macrocell to support two read operations in a cycle. The energy spent on d-out bitlines & mergers in register file is 2.46x than that in RAM, which is higher than expected (expected double). The reason is more connection cells are switched in register file for propagating output data in d-out bitlines. The energy increase in read wordlines is ~3x, which has the same ratio as the complexity increase.

Figure 5.12 on page 69 shows the percentage of dynamic and stand-by energy in total energy consumption. Same as RAM, a write operation requires more junction switches than a read operation. As the capacity grows, the percentage of stand-by energy on total energy is increase.

Table 5.3 on page 70 shows the stand-by energy of register file in different test cases. A "no op" test is implemented to collect stand-by energy when all junctions are inactive.

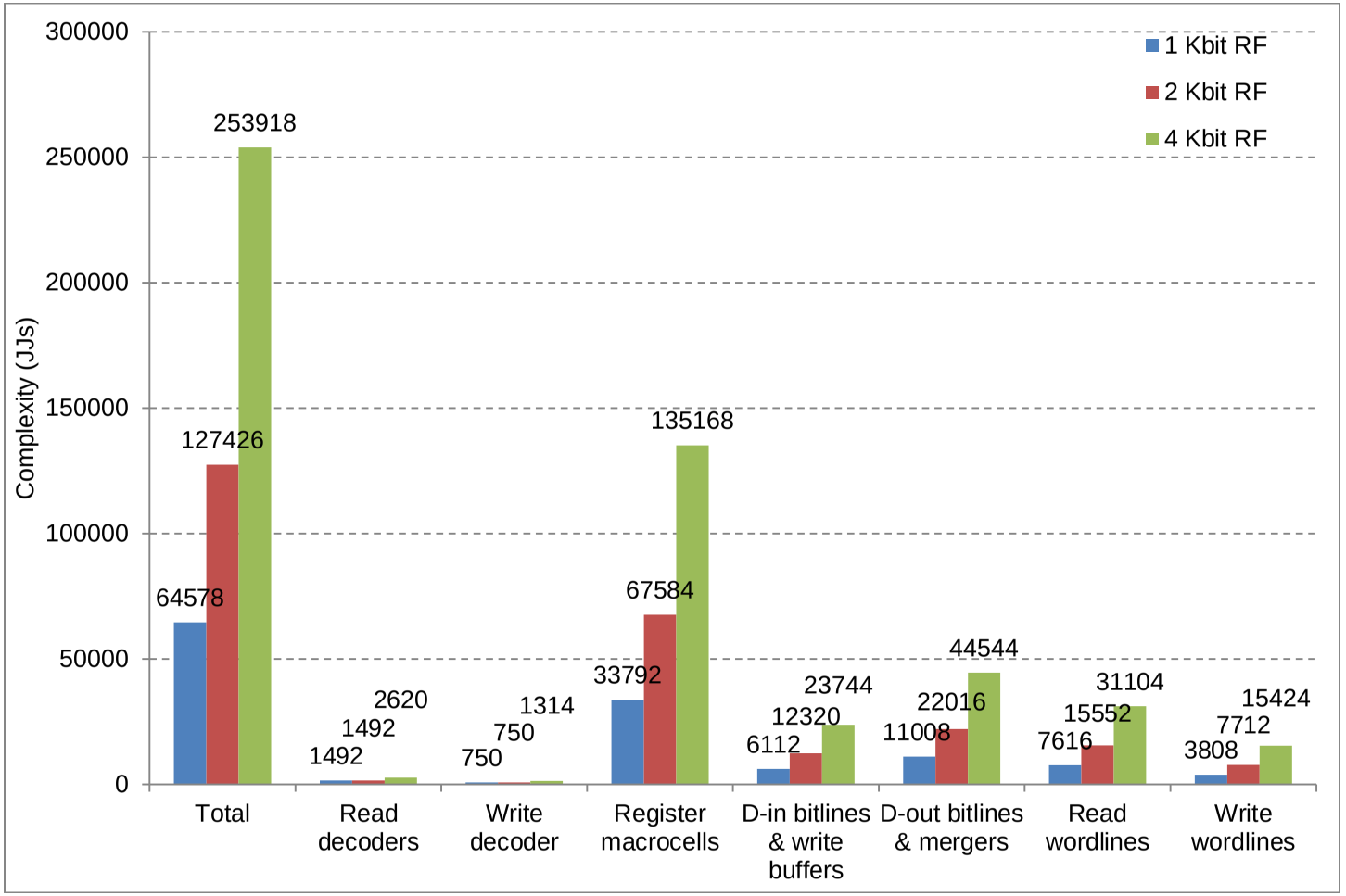


Figure 5.6: Register file complexity breakdown.

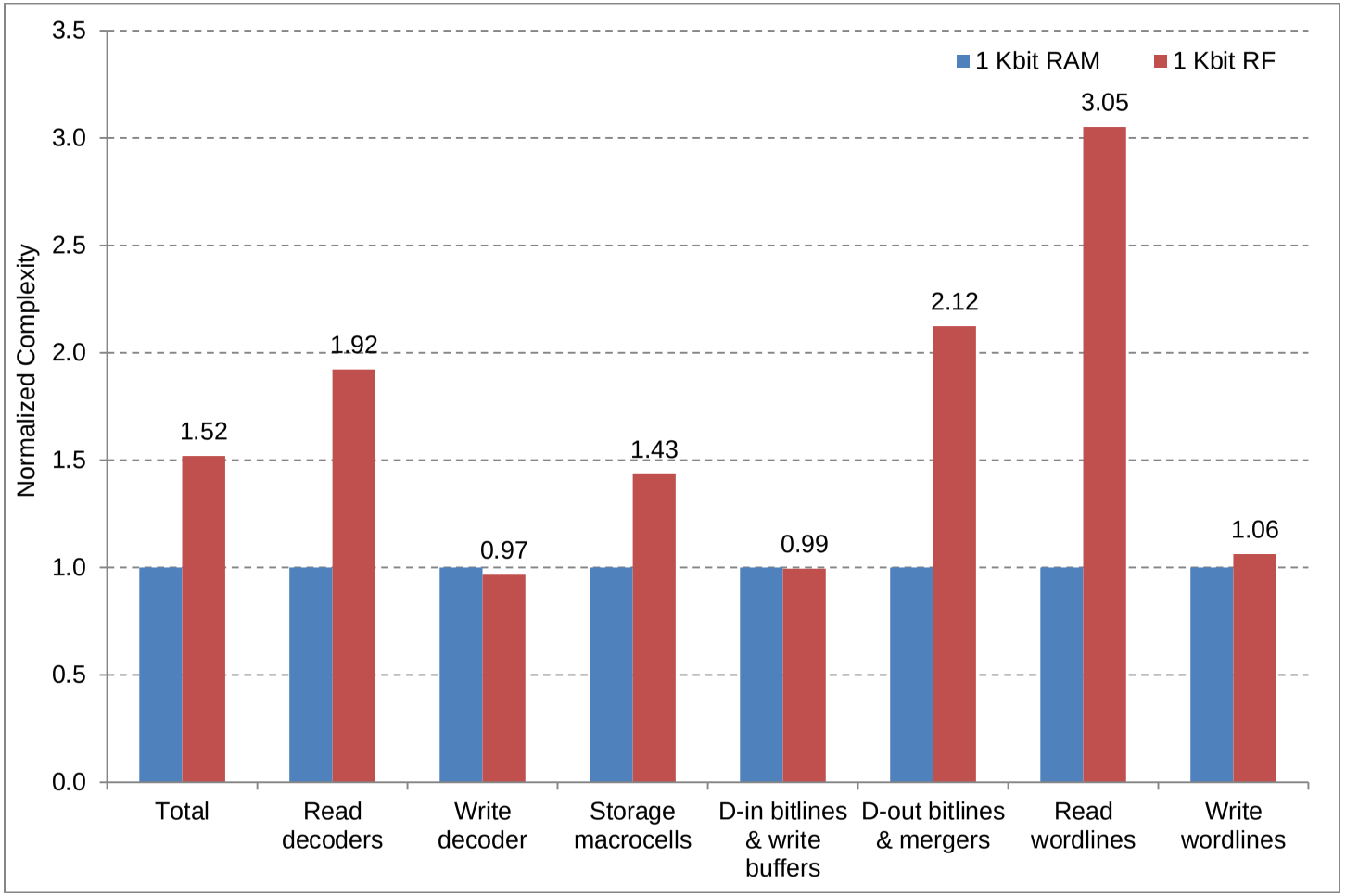


Figure 5.7: Relative complexity of a 1 Kbit register file compared to a 1 Kbit RAM.

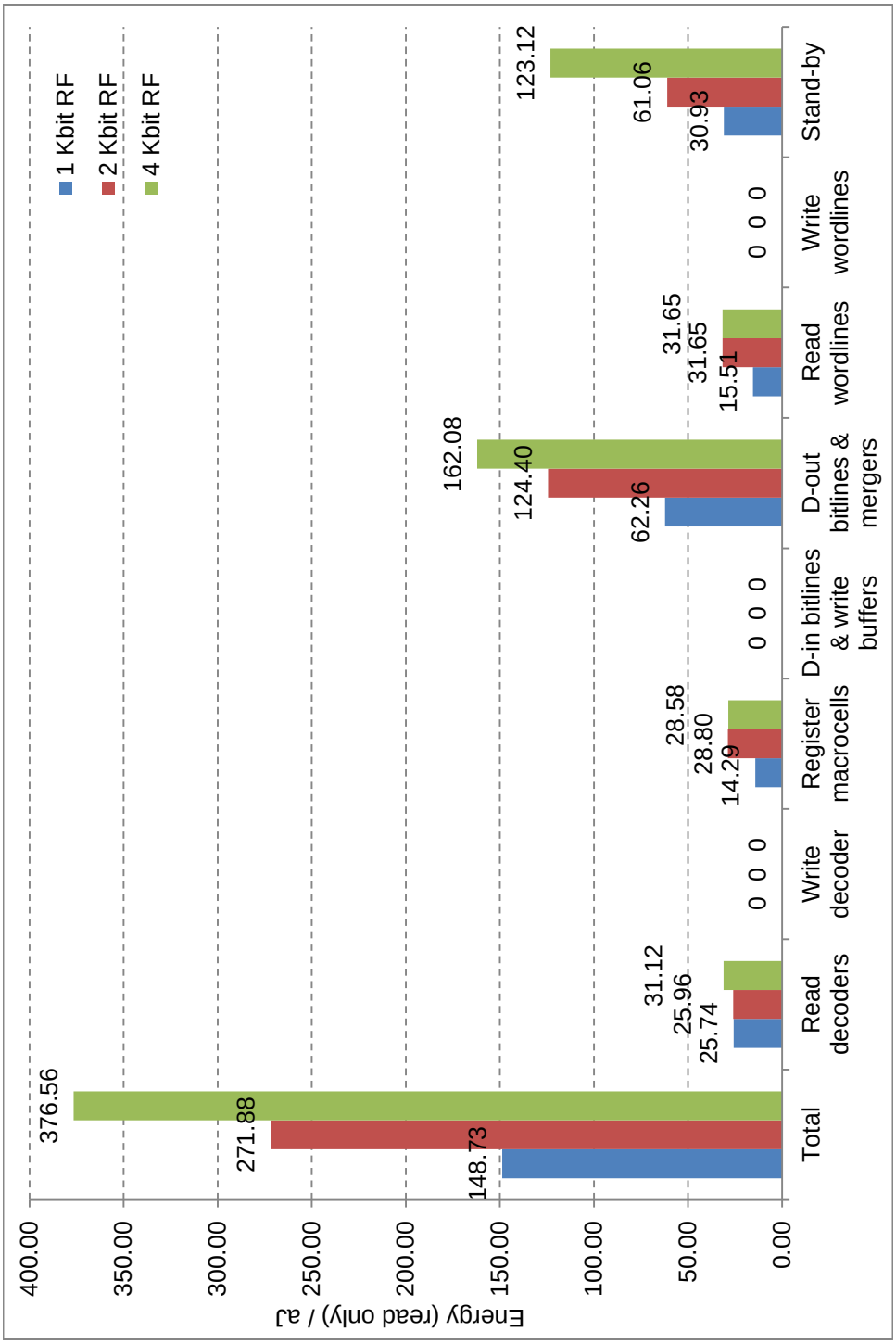


Figure 5.8: Register file energy per read only operation.

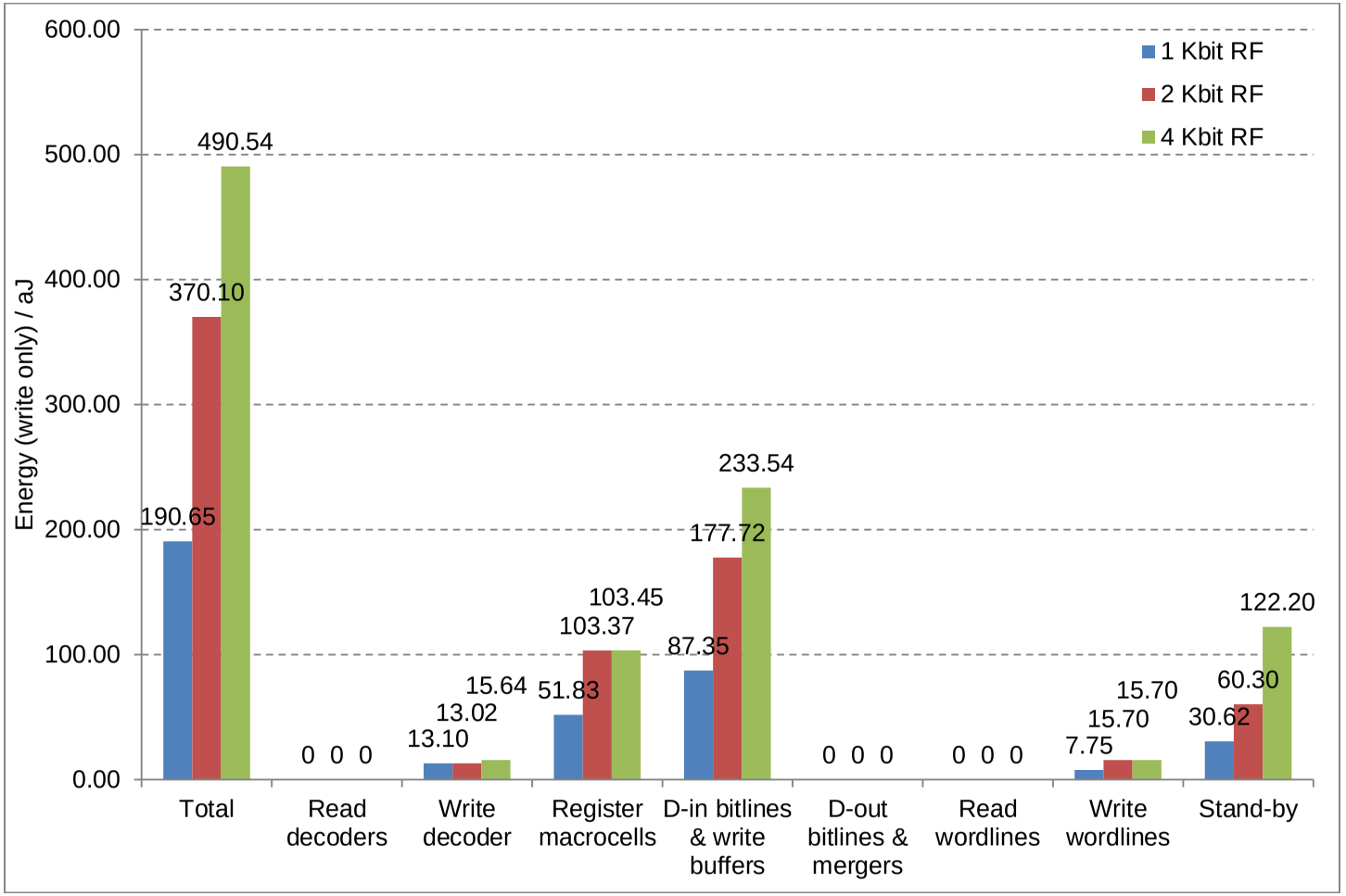


Figure 5.9: Register file energy per write only operation.

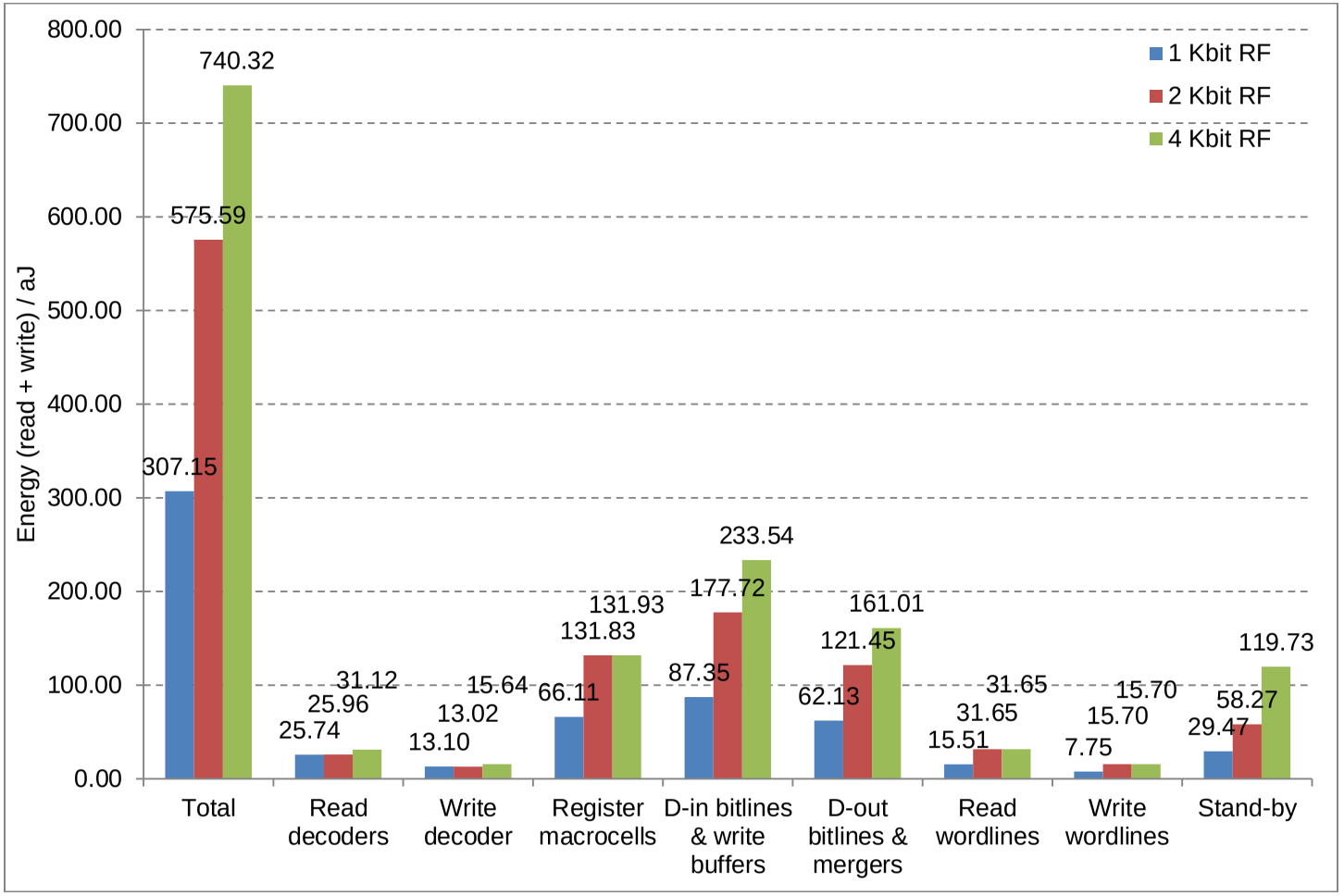


Figure 5.10: Register file energy per read + write operation.

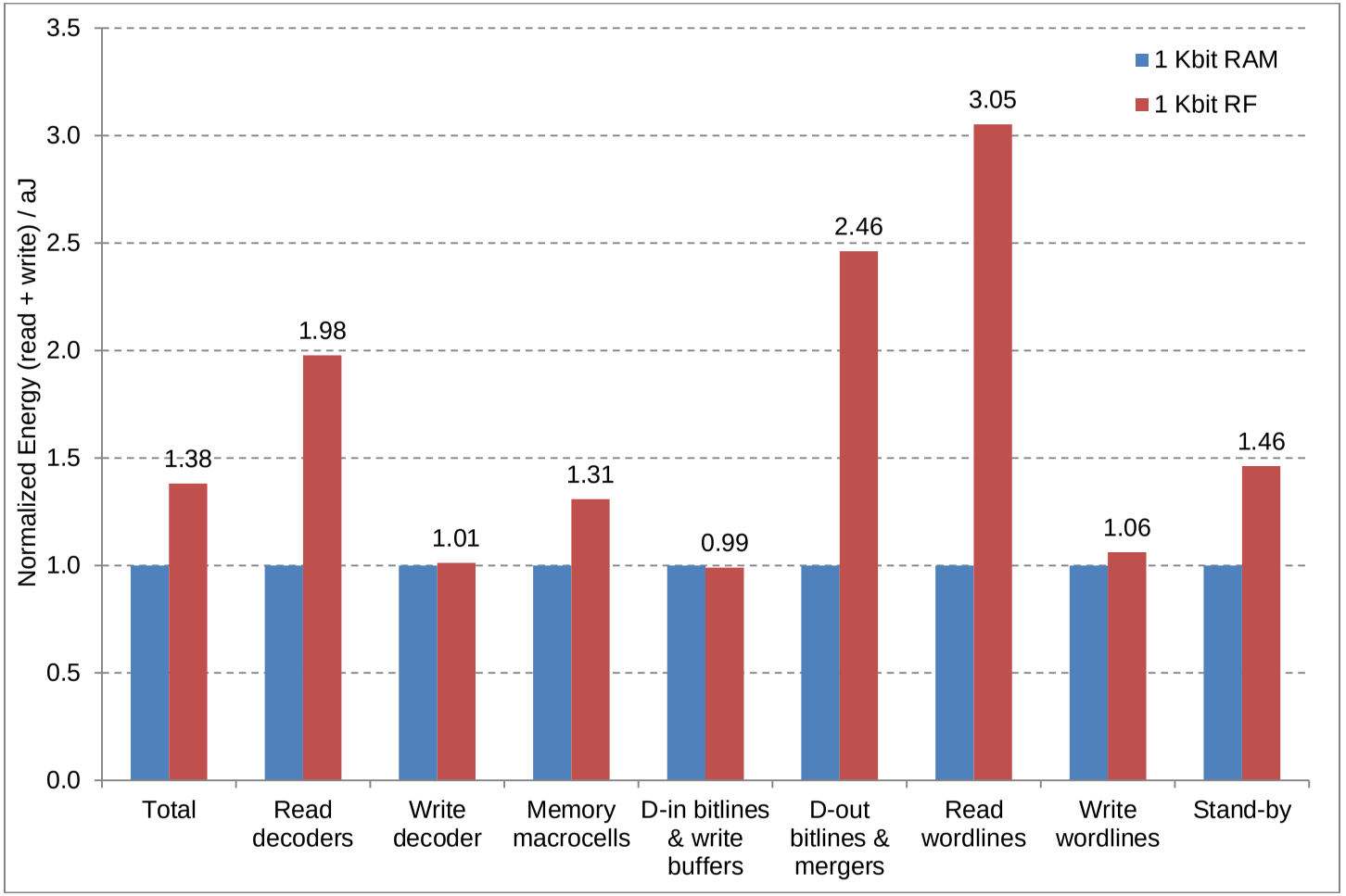


Figure 5.11: Relative energy of a 1 Kbit register file compared to a 1 Kbit RAM.

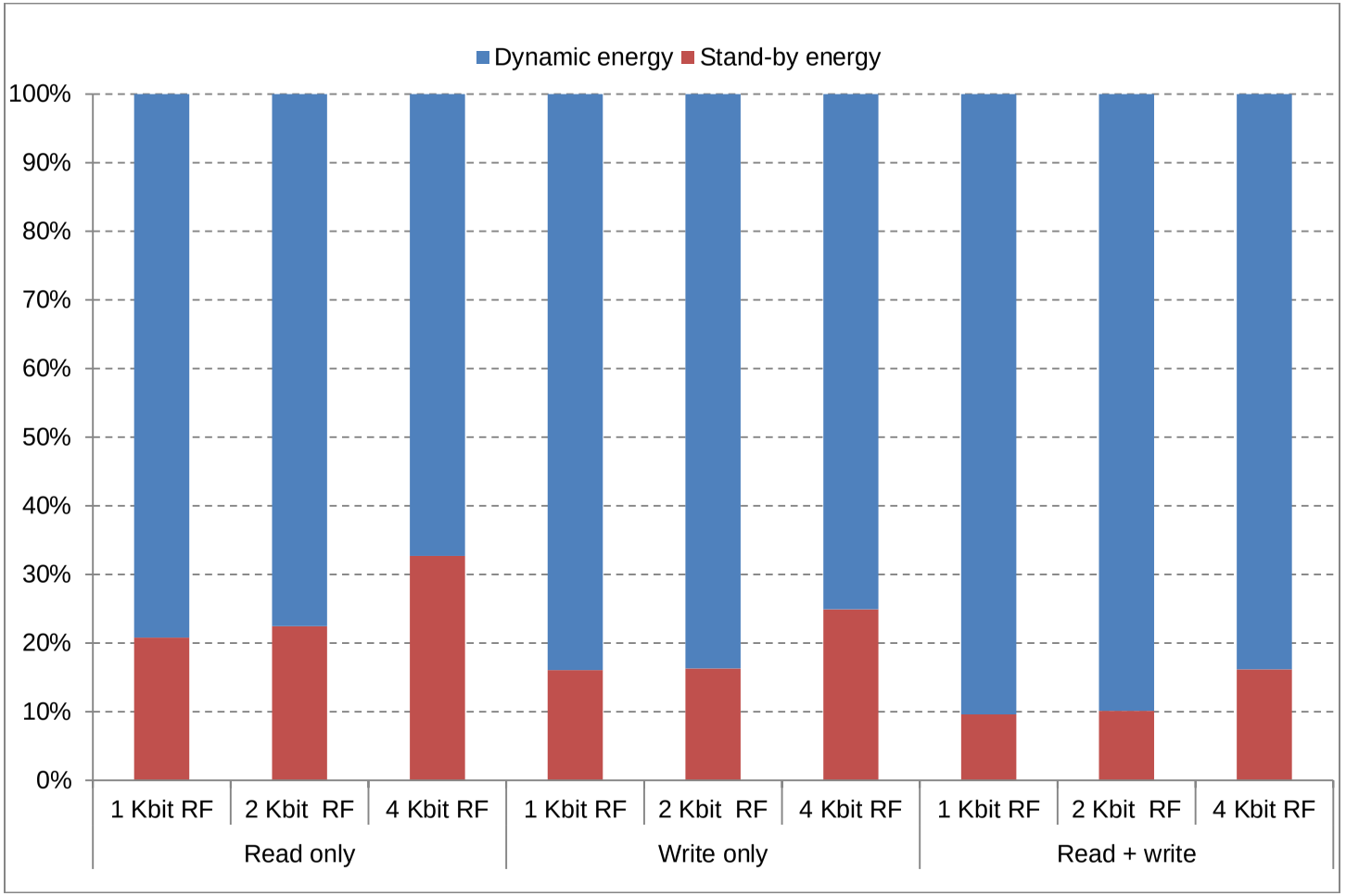


Figure 5.12: Register file dynamic and stand-by energy breakdown.

Table 5.3: Register file stand-by energy.

Test case	**1 Kbit Register file**		**2 Kbit Register file**		**4 Kbit Register file**	
	Stand-by energy (aJ)	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**
Read only	30.93	96.41%	61.06	96.73%	123.12	98.02%
Write only	30.62	95.45%	60.30	95.52%	122.20	97.29%
Read + write	29.47	91.86%	58.27	92.30%	119.73	95.31%
No op	32.09	100.00%	63.12	100.00%	125.61	100.00%

Chapter 6

Write-through and Write-back Caches

Outline

6.1 Design Overview

The efficient memory hierarchy for superconductor RQL processors has to be built with multiple levels of on- and off-chip caches. Our work was focused on Level 1 (L1) 32-bit addressed 32- and 64-bit on-chip data caches implemented with RQL NDRO storage cells. Other (the lower L2 and L3) levels of the cache hierarchy are expected to be implemented with JJ-MRAM technology that can provide much higher storage density [59, 60, 61].

JJ-MRAM has some intrinsic performance issues, such as the order of magnitude difference in the latency/rate of its read and write operations. Because of that, it is not well suited for L1 data caches that need to provide high data throughput and low latency for both read and write operations.

A complete L1 RQL cache design requires a fully-specified interface with a L2 JJ-MRAM cache, which is not available yet. Therefore, we focused on read and write hit cases for L1 RQL caches. The design and analysis of control and interface circuits dealing with L1 misses are left for the future work.

When working on RQL cache design, we reused the efficient techniques developed for our RQL memory design, such as predecoding, binary tree-like broadcasting circuits, separate hierarchical input and output bitlines, and a use of 1-read + 1-write port NDRO-based memory macrocells.

We implemented write-through (WT) and write-back (WB) caches with one read and one write ports. Each type of the cache (WT and WB) has two versions: a 2 Kbit with 32-bit data width and a 4 Kbit with 64-bit data width. We assume that a processor can generate one memory access (load/store) operation and accordingly a cache can perform one (read/write) operation per cycle.

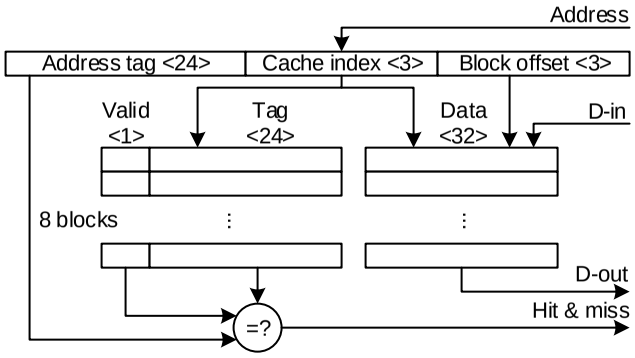
6.2 RQL Cache Design

As shown in Figure 6.1 on page 73, all caches designed in this chapter have 8 cache blocks. For a 32-bit data width version, each block has eight 32-bit, and a cache directory contains a 24-bit tag, a valid bit, and, for the WB cache, a dirty bit indicating whether the data in the corresponding block are changed. For a 64-bit data width version, there are eight 64-bit words in a block and the tag width is 23-bit.

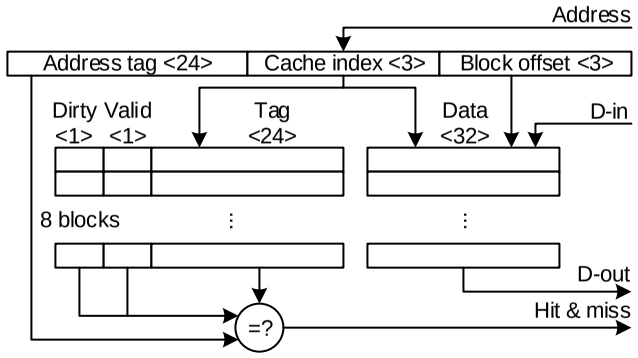
The top-level structure of a 2 Kbit WT and WB cache is shown in Figure 6.2a on page 74 and Figure 6.2b on page 74. There are four major components in a WT cache: decoders, directory, write buffer and data blocks. In a WB cache, a forwarding unit is added. The details of these components will be discussed later.

When executing read operations, both tag and data are read in parallel. Based on the result of tag comparison (hit or miss), the output data from cache block are either used or discarded.

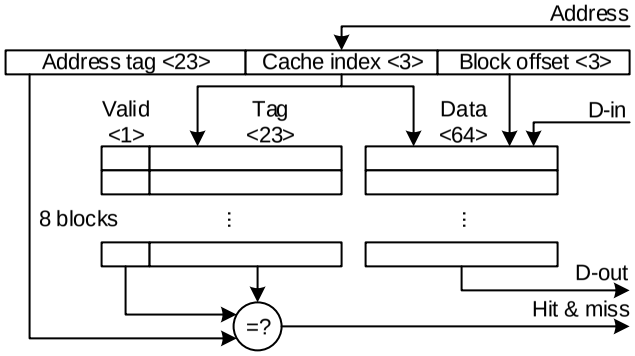
A store (write) operation reads the corresponding L1 cache directory entry first, and, if it is a hit, writes a data word, and also, for the write-back cache, sets the block's dirty bit to '1'. These two steps of write operations are pipelined, with the second, data write, step done in parallel with a read tag step of another load/store operation in the next cycle. A write buffer is used



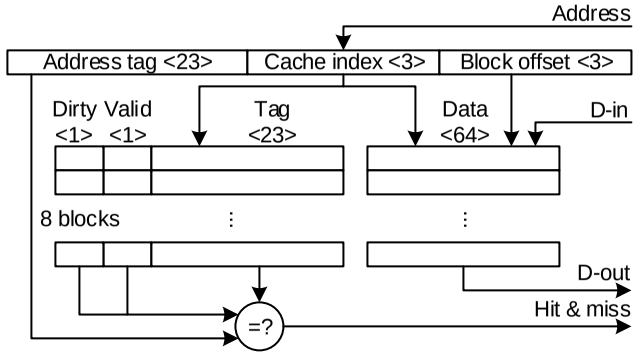
(a) 2 Kbit WT cache.



(b) 2 Kbit WB cache.

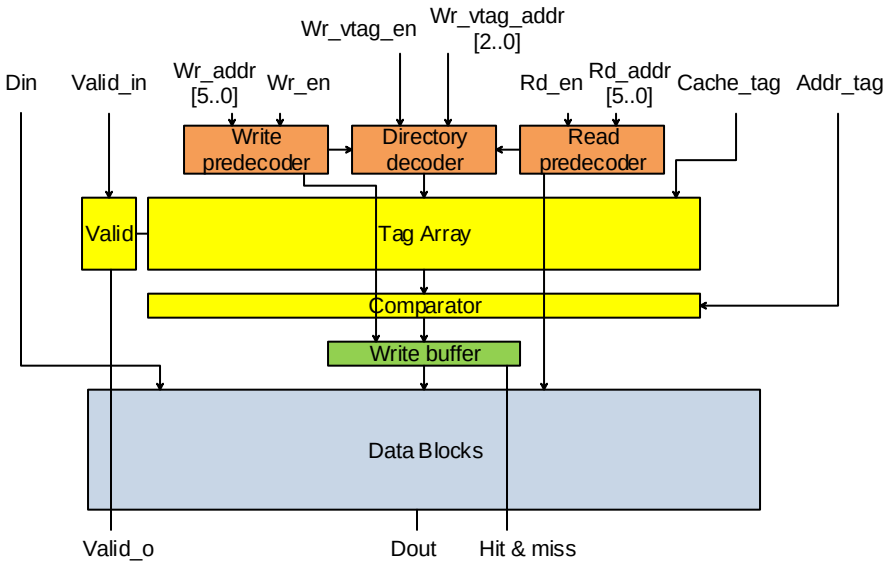


(c) 4 Kbit WT cache.

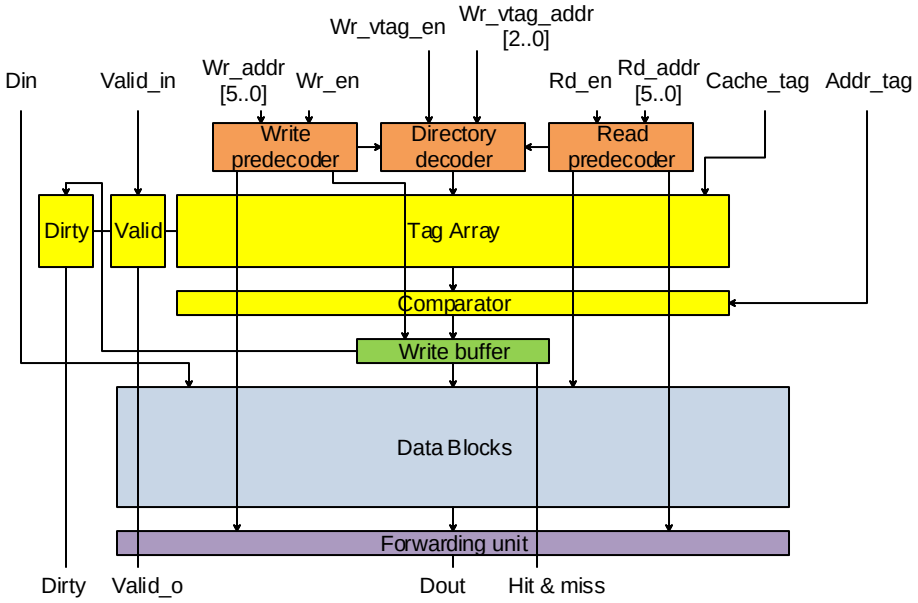


(d) 4 Kbit WB cache.

Figure 6.1: Cache organization.



(a) 2 Kbit WT cache.



(b) 2 Kbit WB cache.

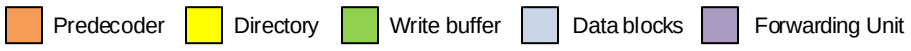


Figure 6.2: Cache top-level structure.

to hold the control signals and data during the time required to determine whether it's a write hit or miss.

The latency of cache operations is more than one cycle. To provide a throughput of 1 read/write operation per cycle, cache operations are pipelined, so several read / write operations can be at different stages of execution at any given cycle.

The no-write allocate [62] policy is used to deal with write misses, which means a miss from write operation does not affect the L1 cache. The write operation that has a miss in the L1 cache is directly sent to the lower level cache.

Compared to a write-back cache, a write-through cache has lower complexity, but requires more write operations to the lower level cache. As the result, a write-through cache is designed as a L1 instruction cache, which has less write operations and more read operations. A write-back cache is designed for a L1 data cache with frequent write operations requiring an efficient write back policy to decrease the write traffic to the L2 cache.

6.2.1 Decoders

There are three decoders in a cache: a read decoder, a write decoder and a directory decoder. Same as the decoders in RAM and register file, both read and write decoders use predecoding [63] technology, where the higher 3 bits of the read or write address (3-bit cache index) is predecoded into 8-bit one-hot block select signal, and the lower 3 bits of the read or write address (3-bit block offset) is predecoded into 8-bit one-hot word select signal. In the final stage decoder (in the middle of the data blocks), these signals are used to generate word select signals that are sent through wordlines to select the rows.

Directory decoder is used to decode read or write cache index to control directory. During a cache read or cache write operation, the higher 3 bits of the read or write address (3-bit cache index) in the read or write decoders also be sent to the directory decoder as a 3-bit read address. After decoding, 8-bit one-hot read directory signal is generated to read tag, valid bit and dirty bit (in WB cache) from one of the eight entries. During a tag write operation, 3-bit directory write address will be decoded to 8-bit one-hot write directory signal to select the entry be written.

6.2.2 Directory

For a 2 Kbit WT cache, directory has three components: a tag array, a valid bit column, and a tag comparator. Cache tags are stored in an 8×24-bit tag array. This tag array consists of six 8×4-bit NDRO-based cell arrays which

are described in Chapter 4. Valid bit column is an 8 × 1-bit NDRO-based cell column used to store valid bit for each entry. The tag comparator is used to compare 24-bit cache tag with 24-bit address tag bit by bit using XOR gate arrays then generate hit or miss signal.

For a 4 Kbit cache, the width of a tag is 23-bit. The tag array contains five 8 × 4-bit NDRO-based cell arrays and an 8 × 3-bit NDRO-based cell array. An 8 × 3-bit NDRO-based cell array is achieved by cutting 1-bit NDRO-based cell column from an 8 × 4-bit cell array.

In a WB cache, there is an 8 × 1-bit NDRO-based dirty bit column in the directory. Corresponding bit can be set to '1' by write hit signal and reset to '0' by a directory write operation.

6.2.3 Write Buffer

This is the buffer used to hold the control and data signal until the hit or miss is resolved during a write operation. The signals been buffered here are 8-bit one-hot block select signal, 8-bit one-hot word select signal and input data. If the result of a write operation is a hit, a write hit signal unblocks these signals to complete the write operation. If it is a write miss, these signals would be discarded.

6.2.4 Data Block

The data blocks in cache are employed from data slice in RAM. For 2 Kbit cache, there are eight 8-word × 32-bit data blocks, and each data block has eight 8-word × 4-bit data arrays. For 4 Kbit cache, there are eight 8-word × 64-bit data blocks, and each data block has sixteen 8-word × 4-bit data arrays.

6.2.5 Forwarding Unit

The forwarding unit is used to resolve the read-after write data hazard: if there is a write hit follow by a read hit to the same destination, the read access cannot get the latest data written by the write operation.

To resolve this issue, a data forwarding unit is placed within the write-back data cache to guarantee that the read operation that immediately follows a write operation with the same address always gets the correct output value by forwarding it directly from the write operation.

The forwarding unit buffers the input data, 3-bit cache index, 3-bit block offset, write hit and read hit signal. If the previous operation is write hit, the following operation is read hit, cache index and block offset of two operations

are the same, we can guarantee that these two operations are target to the same address. In this case, a multiplexer would select the input data buffered in the forwarding unit as the valid output data and send to the output.

6.2.6 Pipeline

In order to improve the throughput, pipelining is employed in the designs, as shown in Figure 6.3 on page 77. For a directory check, there are 3 stages: decode, tag read and compare. For a block access, there are also 3 stages: predecode, final decode and data read/write. During a read operation, both the directory and data section are read in parallel. During a write operation, the block write will stall after finished predecode stage until the hit/miss is resolved by the directory check.

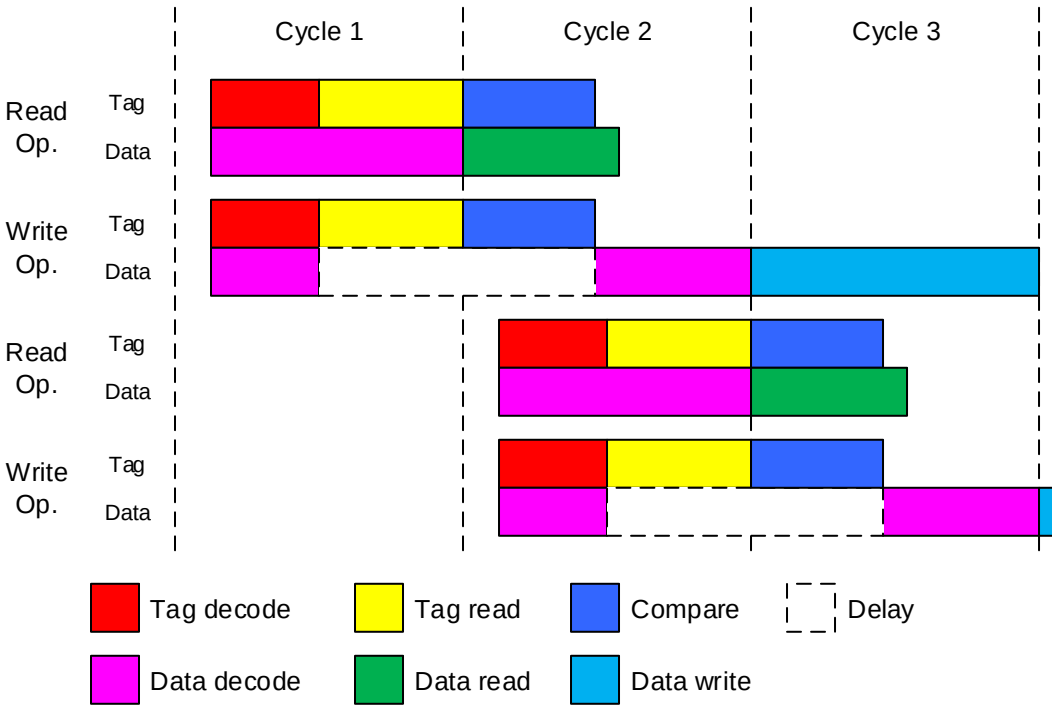


Figure 6.3: Cache pipelines (not incl. clock skew).

6.3 Simulation Results and Discussion

In this section, we evaluate the scaling of two types of caches (write-through and write back) as the capacity grows (from 2 Kbit to 4 Kbit). Simulation summary is shown in Table 6.1 on page 78.

Random test vectors are used to verify the functionality of these designs. All the statistics are collected with the minimum critical current of 38 µA at the temperature of 4.2 K.

Table 6.1: Summary of the cache designs.

Cache type	Write-through		Write-back	
Data capacity, Kbits	2	4	2	4
Data width, bits	32	64	32	64
Depth, words	64	64	64	64
Tag width, bits	24	23	24	23
Valid bit	Yes	Yes	Yes	Yes
Dirty bit	No	No	Yes	Yes
Clock frequency, GHz	8.5	8.5	8.5	8.5
Complexity, JJs	93394	177136	94986	179686
Read latency (incl. clock skew), ps	293.60	323.10	308.11	337.61
Write latency (incl. clock skew), ps	439.45	468.95	439.45	468.95
Read hit (incl. clock skew), ps	182.60	182.60	182.60	182.60
Write hit (incl. clock skew), ps	199.00	199.00	201.76	201.76
Average energy/op, aJ	218.26	372.44	233.50	395.65

6.3.1 Latency

As the capacity grows from 2 Kbit to 4 Kbit, the read and write latencies are increase by one quarter clock cycle (one phase). Same as RAM and register file, this is result from additional time required to send signal over a 64-bit row.

The read latency is increase when comparing WB cache with WT cache. This difference comes from the latency of forwarding unit in a WB cache.

The read hit and write hit latency is defined as the time required to generate read hit or write hit signal, respectively. The write hit latency larger in WB cache than WT cache. This results from the extra time to split the write hit signal, which should be send to dirty bit column as a control signal.

6.3.2 Design Complexity

Figure 6.4 on page 80 shows the complexity breakdown of four caches. As the capacity doubles, the complexity of decoders and write buffer remains unchanged. The complexity of directory is reduced because tag width in a 4 Kbit cache is reduced by one bit compared to a 2 Kbit cache. The complexity of the data blocks is approximately doubled as expected.

A directory in a WB cache requires more junctions than a directory in a WT cache because of a dirty bit column in a WB cache. The complexity of decoders and write buffer in a WB cache are also increase because additional interconnect or control circuits are added to support this dirty bit column.

6.3.3 Energy Consumption

Because lower level memories (such as L2 cache) are not available, we only study cases when it is a read hit or write hit. There are three test cases: read hit, write hit and no op. Initially, the tag array and data blocks are filled with random data. Valid bits are set to '1' and dirty bits in the WB cache are set to '0'. In a read hit, a testbench provides random 6-bit read address to a cache. To ensure a read hit, this testbench also provides address tags that is the same as the random cache tags in the directory. In a write hit, the testbench provides random 6-bit write address, random input data and address tags that is the same as cache tags in the directory. During a no op case, neither read nor write operation is provided. The energy consumption and breakdown is shown in Figure 6.5 on page 81 and Figure 6.6 on page 82.

As the capacity doubles, the energy spend on data blocks are doubled as expected. The energy spend on directory is less because tag width is reduced in a 4 Kbit cache. Forwarding unit consumes a small amount of energy to buffer the read address, write address and input data in a WB cache.

By comparing WT cache with WB cache, decoders and directory consumes more energy in a WB cache to support dirty bit column and additional interconnect and control circuits.

Figure 6.7 on page 83 shows the percentage of dynamic and stand-by energy in total energy consumption. A write cache operation requires more junction switches than a read cache operation. As the capacity grows, the percentage of stand-by energy is reduced. As the capacity grows, the percentage of stand-by energy grows.

Table 6.2 on page 84 shows the stand-by energy collected in the no op case where all junctions are inactive. By comparing with read hit and write hit, we can see that only a small amount of junctions are switched during an operation.

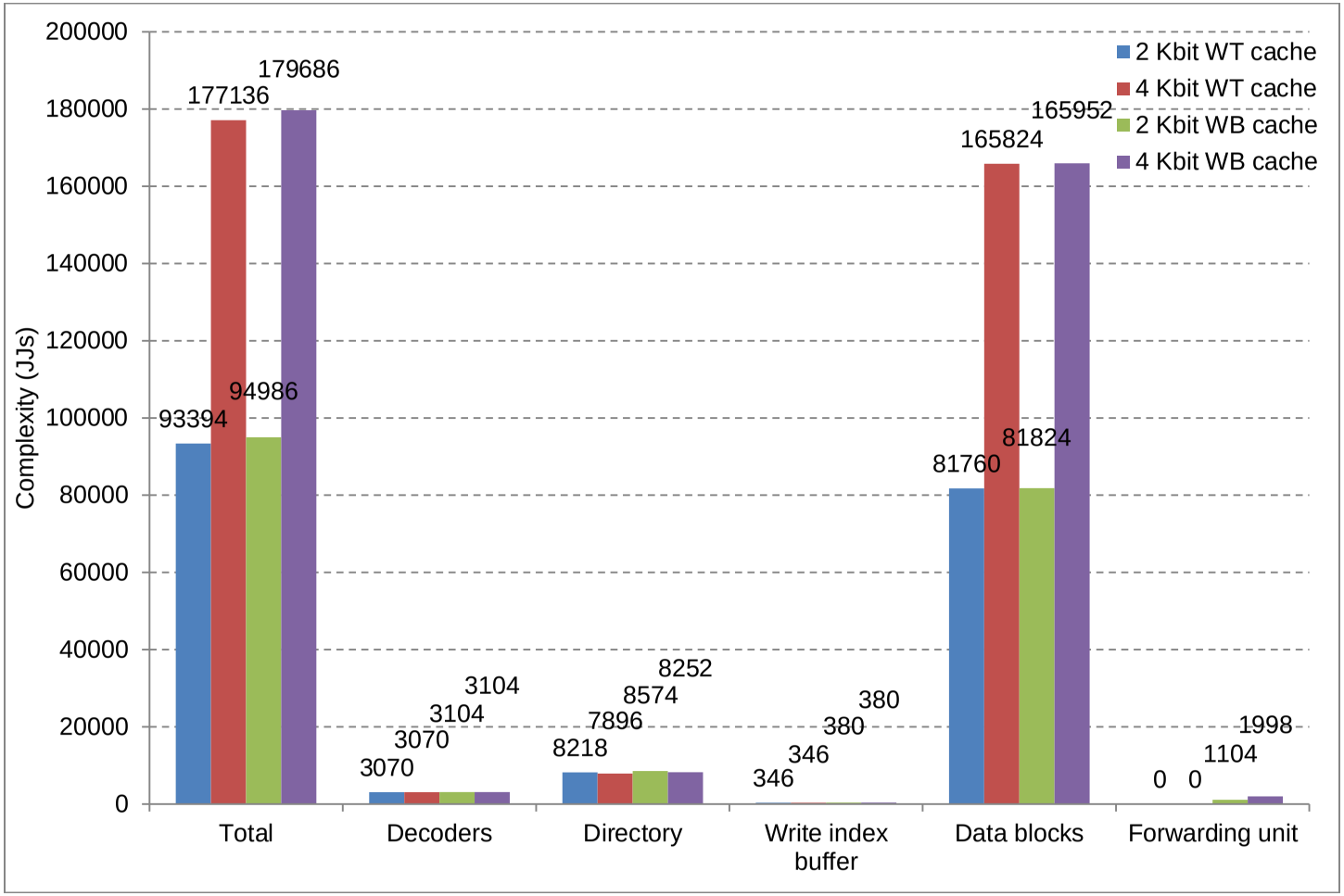


Figure 6.4: Cache complexity breakdown.

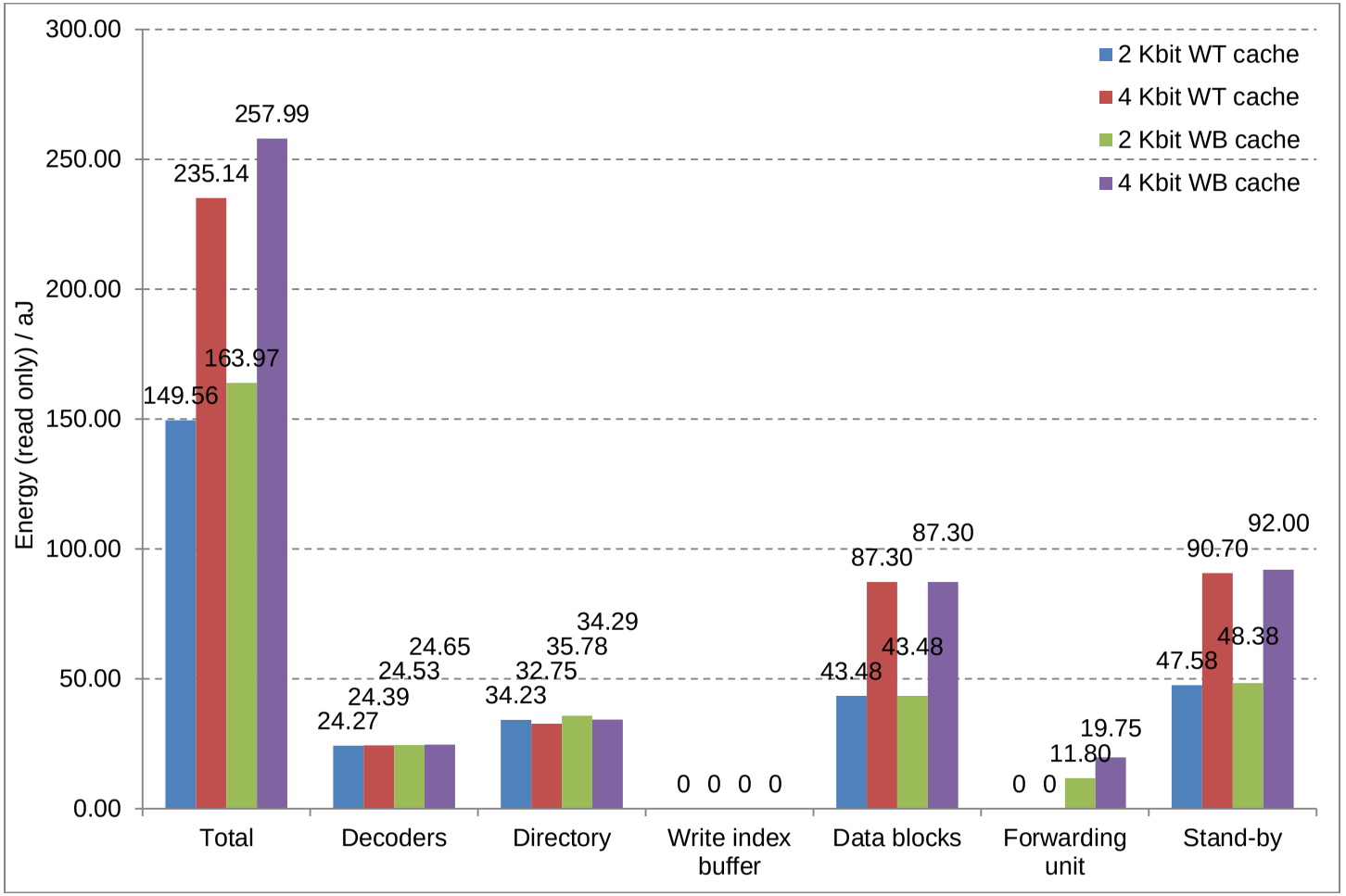


Figure 6.5: Cache energy per read hit operation.

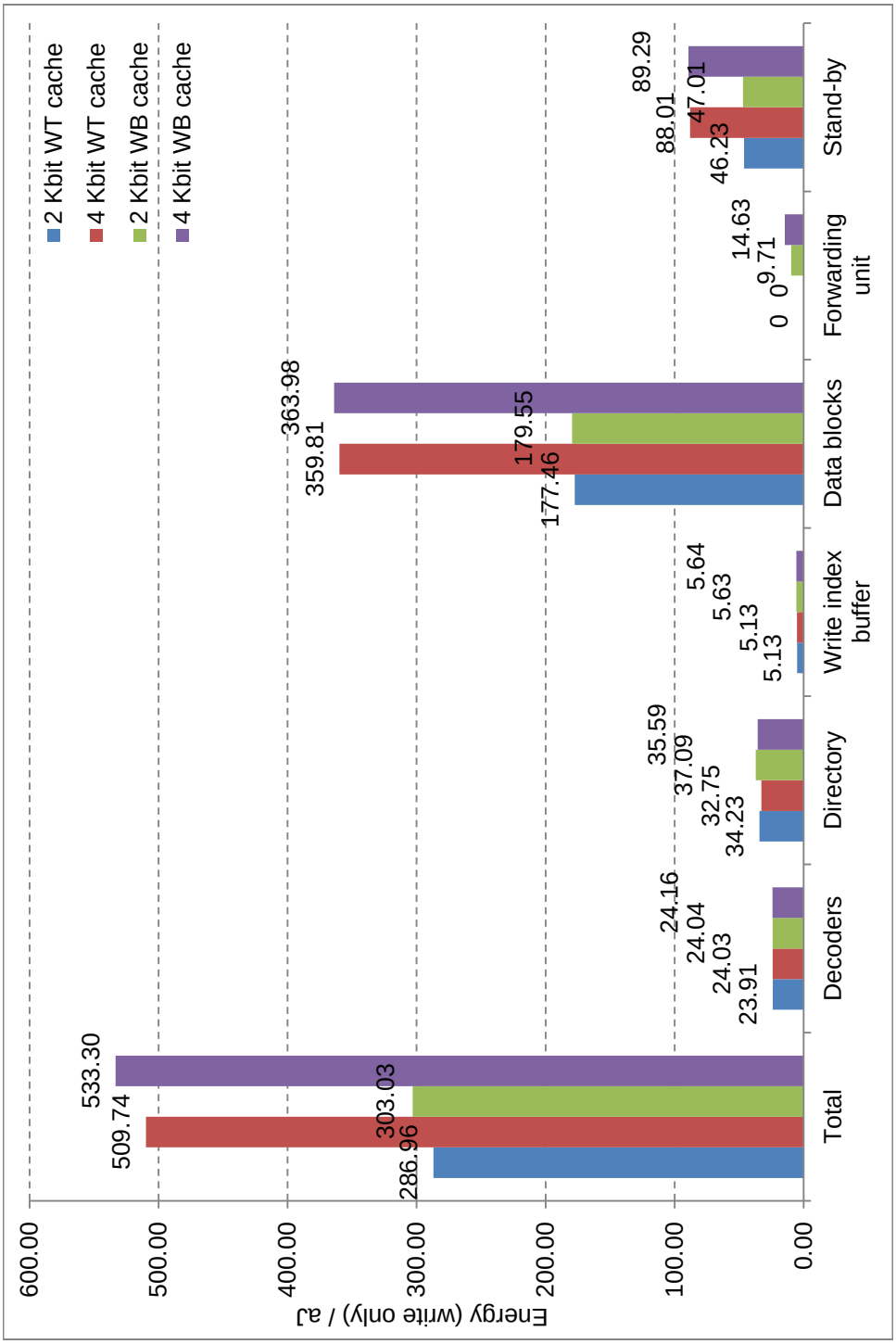


Figure 6.6: Cache energy per write hit operation.

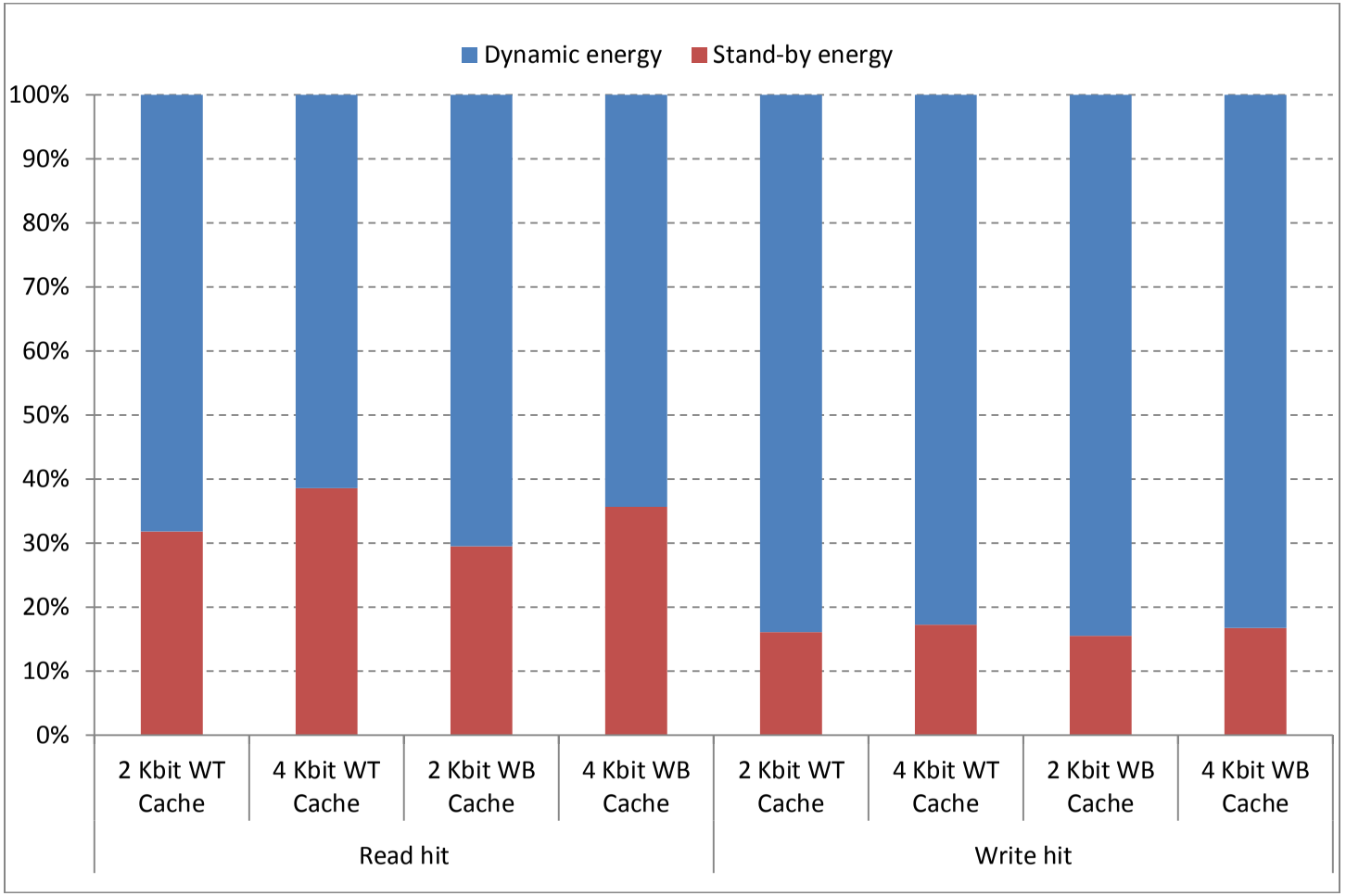


Figure 6.7: Cache dynamic and stand-by energy breakdown.

Table 6.2: Cache stand-by energy.

Test case	**2 Kbit WT Cache**		**4 Kbit WT Cache**		**2 Kbit WB Cache**		**4 Kbit WB Cache**	
	Stand-by energy (aJ)	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**
Read hit	47.58	97.96%	90.70	98.47%	48.38	97.75%	92.00	98.29%
Write hit	46.23	95.19%	88.01	95.56%	47.01	94.98%	89.29	95.39%
No op	48.57	100.00%	92.11	100.00%	49.49	100.00%	93.60	100.00%

Chapter 7

First-In First-Out Buffers

Outline

7.1 Design Overview

We designed and studied several first-in first-out (FIFO) buffers, namely: a 128 bit (4-word × 32-bit), a 256 bit (8-word × 32-bit), and a 512 bit (8-word × 64-bit) ones. These FIFO buffers have one read port and one write port and can perform two (one read and one write) operations per cycle. A new shift-register based decoder was developed to reduce the complexity of the control circuit and the latency. With this new decoder, the read latency is lower than one clock cycle for a 32-bit design at the frequency of 8.5 GHz.

7.2 RQL FIFO Buffer Design

There are two parts in a FIFO buffer: a memory array and a control circuit that includes read and write address pointers. The memory array design employs the techniques used in our RAM and cache designs, such as binary tree-like broadcasting circuits, separate bitlines, and 1-read and 1-write port NDRO memory macrocells discussed in Chapter 4. Based on the value of the read and write address pointers, the FIFO control circuit generates signals to read / write relevant data words. It also updates the pointers and, when necessary, sets full and empty flags.

Top-level schematic of a 256 bit FIFO is shown in Figure 7.1 on page 86.

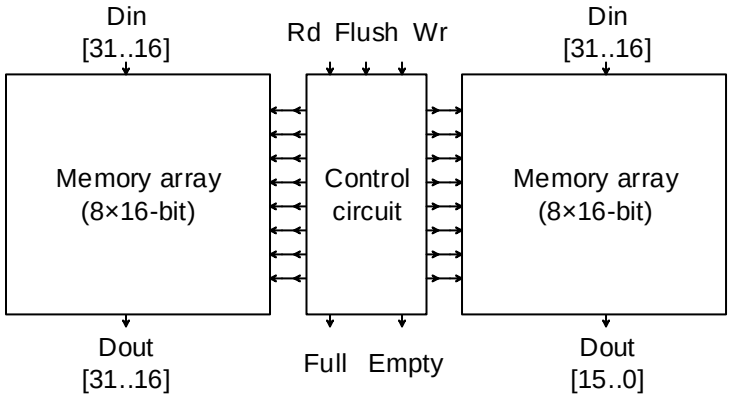


Figure 7.1: Top-level structure of a 256 bit FIFO.

7.2.1 Control Circuit

We explored and rejected the use of counters for the FIFO control circuit implementation because of its high complexity and latency. Instead, our FIFO control circuit consists of two 1-bit wide circular shift registers (one bit/word) built with NDRO2 cells. These shift registers implement one-hot encoded read and write address pointers. The logical '1' in the pointer indicates the row (word) to be read or written in the next read or write operation, respectively. A comparator is used to generate full and empty flags based on the FIFO operation and values of the read and write address pointers, as shown in Figure 7.2 on page 87.

To reduce access latency, the shift registers are placed vertically in the middle of the memory array (one bit per row). The row select signals generated from the shift registers are broadcasted across the corresponding rows. The comparator is placed between the read and write shift registers.

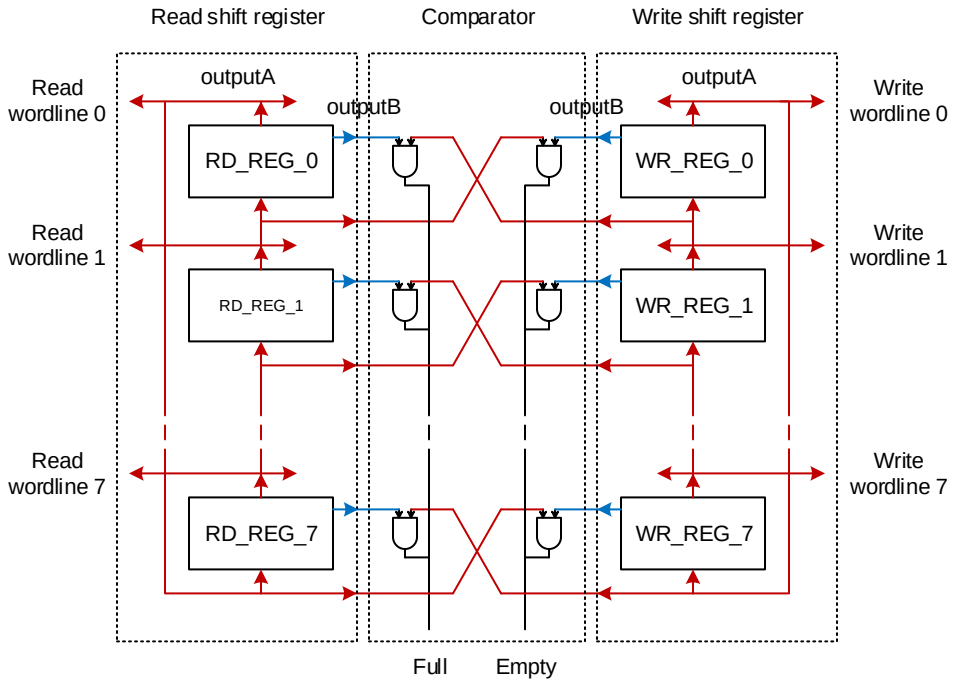


Figure 7.2: The FIFO control circuit schematic (8-word version).

The direction of the shift register is from bottom to top. The output of the top bit is connected to the input of the bottom bit. A reset operation sets an initial value of each of the pointers with a non-zero bit in the top bit. The reason for this direction is the clock skew and propagation delay in RQL circuit. If the direction were set in the opposite way, the critical path would be from the bottom bit to the top bit. The signal cannot be transmitted through this path within a quarter cycle (one phase) because of the clock skew and data propagation delay.

The NDRO2 cell has two read ports, namely, outputA and outputB. The read pointer can be read from outputA when there is a read operation; the write pointer can be read from outputA when there is a write operation. Both read and write points can be read from outputB if there is a read or write operation. The output data from outputA has three destinations: 1) a read / write wordline to select a row, 2) the data input of the next storage element and 3) the comparator for full and empty flags generation. The data from outputB will be sent to the comparator.

The full and empty flags are generated based on: 1) if the read pointer is equal to write pointer and the latest operation is a write, then the FIFO buffer is full, 2) if the read pointer is equal to write pointer and the latest

operation is a read, then the FIFO buffer is empty. As shown in Figure 7.2 on page 87, a write operation will send the write pointer to the and gates on the left to compare with the read pointer, and set a full flag if two pointers are matched; a read operation will send the read pointer to the and gates on the right to compare with the write pointer, and set an empty flag if two pointers are matched.

7.2.2 Memory Array

The memory array in a FIFO buffer is employed from a RAM data slice that includes 1 read and 1 write ports memory macrocells. The schematic of a memory array is shown in Figure 7.3 on page 88. The memory array only has 4 words or 8 words. Since the memory array only has 4 words or 8 words, global D-in and D-out bitlines, write buffers and mergers are not necessary. Input data are propagated to all the macrocells in a column through d-in bitlines (local d-in bitlines in RAM). Output data are collected by d-out bitlines (local d-out bitlines in RAM). Read wordlines and write wordlines have the same structure as wordlines in RAM.

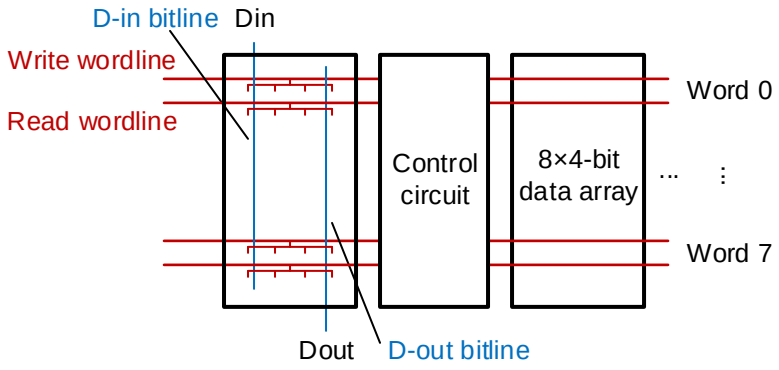
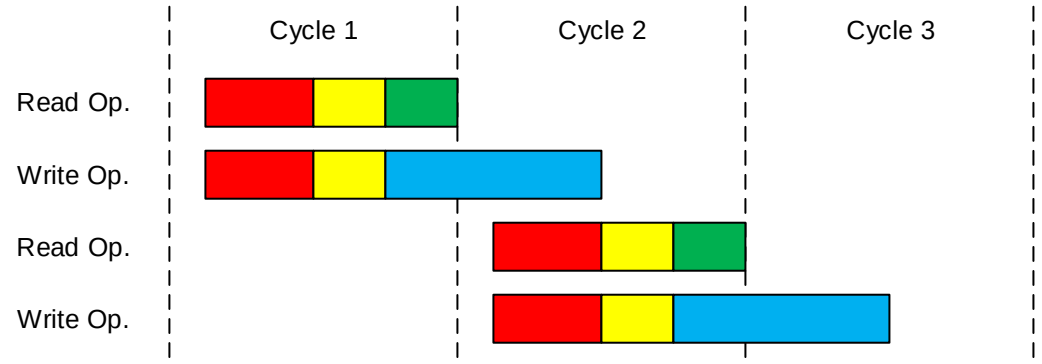


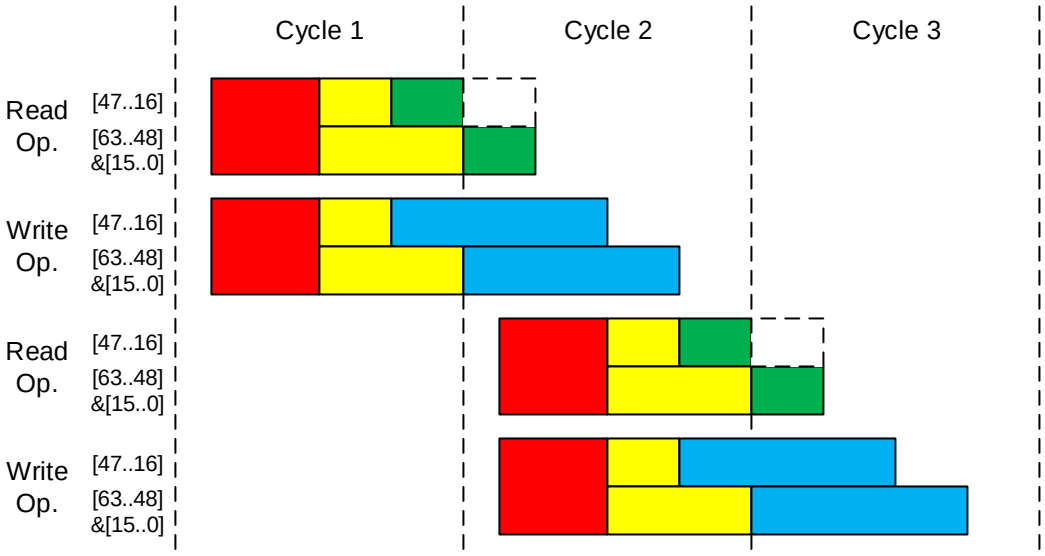
Figure 7.3: The memory array.

7.2.3 Pipeline

All FIFOs are pipelined, as shown in Figure 7.4 on page 89. Same as RAM, for read operation, there are 3 stages: decode, worldline broadcast, and NDRO read & D-out. For write operation, there are 3 stages: decode, wordline broadcast and NDRO write. The FIFO write operations have a slightly longer latency than read operations. Despite that, even in the case of a write operation followed by a read operation, the read operation gets the latest data written by the write operation without any use of additional logic (such as a forwarding



(a) The FIFO pipelines (32-bit data width).



(b) The FIFO pipelines (64-bit data width).

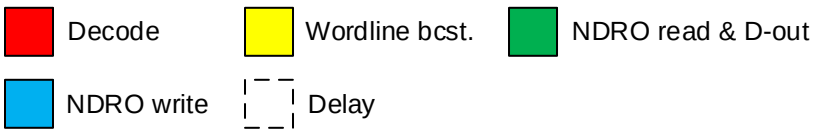


Figure 7.4: The FIFO pipelines (not incl. clock skew).

unit). This is because the write operation is completed one quarter cycle (one phase) earlier than the read operation gets access to the written memory cells.

For 64-bit width version, read and write access to NDRO cells in bit 63 to bit 48 and bit 15 to bit 0 are one quarter cycle (one phase) later than the NDRO cells in bit 47 to bit 16.

7.3 Simulation Results and Discussion

Same as the previous designs, the simulation results of FIFO show the influence on latency, complexity and energy as capacity increase.

The results are summarized in Table 7.1 on page 90. All the statistics are collected with the minimum critical current of 38 µA at the temperature of 4.2 K.

Table 7.1: Summary of the FIFO designs.

Data capacity, bits	128	256	512
Data width, bits	32	32	64
Depth, words	4	8	8
Clock frequency, GHz	8.5	8.5	8.5
Complexity, JJs	5380	10578	20434
Read latency (incl. clock skew), ps	90.85	107.41	138.06
Write latency (incl. clock skew), ps	143.54	153.83	184.50
Full / empty latency (incl. clock skew), ps	76.29	92.85	93.09
Average energy/op, aJ	49.58	72.07	132.43

7.3.1 Latency

As the depth doubles (from 128 bit FIFO to 256 bit FIFO), all latencies are increased because the propagation distance grows. As the width doubles, the read and write latencies are increased by one quarter cycle (one phase), same as RAM and register file. The growth of width does not effect the latency of full / empty flag.

7.3.2 Design Complexity

Figure 7.5 on page 92 shows the complexity of three types of FIFO. As the depth doubles, the complexities of all components are doubled. As the width

doubles, the complexities of all components except decoder are doubled. The complexity of a decoder only depends on the depth (number of words) of a FIFO.

7.3.3 Energy Consumption

Two test cases are applied, namely: read + write and no op. FIFO is initially half full with random data. In read + write case, we perform one read and one write operation per cycle: read data from the head and store new random data to the tail. In no op test case, neither read nor write operation is applied.

The result is shown in Figure 7.6 on page 93. As the depth doubles, the energy spent on decoder, FIFO macrocells, din and dout bitlines are increased. The energy spent on wordlines remain unchanged, because only one set of wordlines are triggered for each operation. As the width doubles, the energy spent on all components expect decoders are doubled.

Figure 7.7 on page 94 shows the percentage of dynamic and stand-by energy of FIFO. The percentage of stand-by energy is very low compared to designs in previous chapters. The reason is that the FIFO capacity is small compared to other designs, which means less stand-by energy is needed to store the data. Table 7.2 on page 95 shows the stand-by energy collected in the no op case where all junctions are inactive.

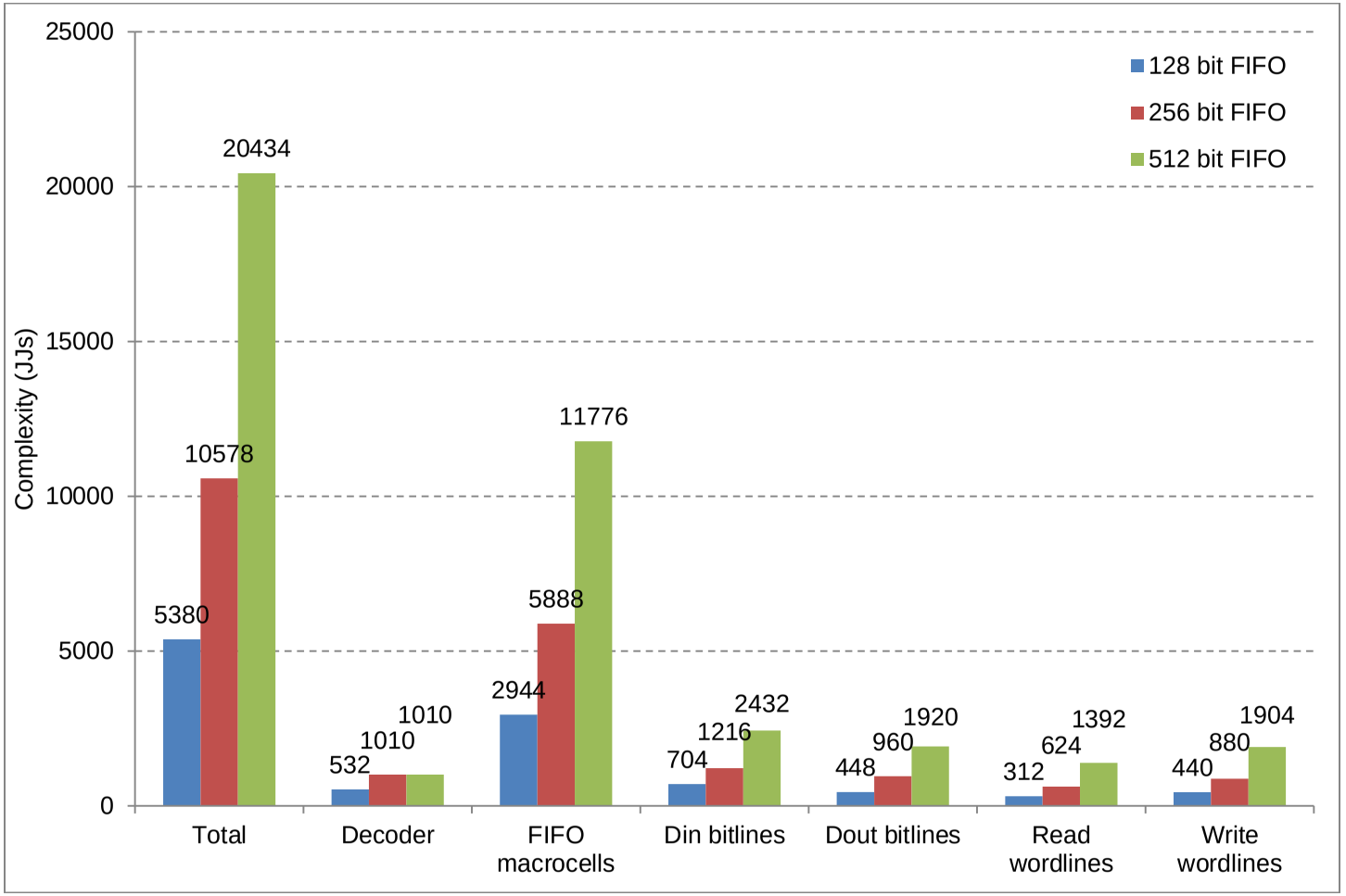


Figure 7.5: FIFO complexity breakdown.

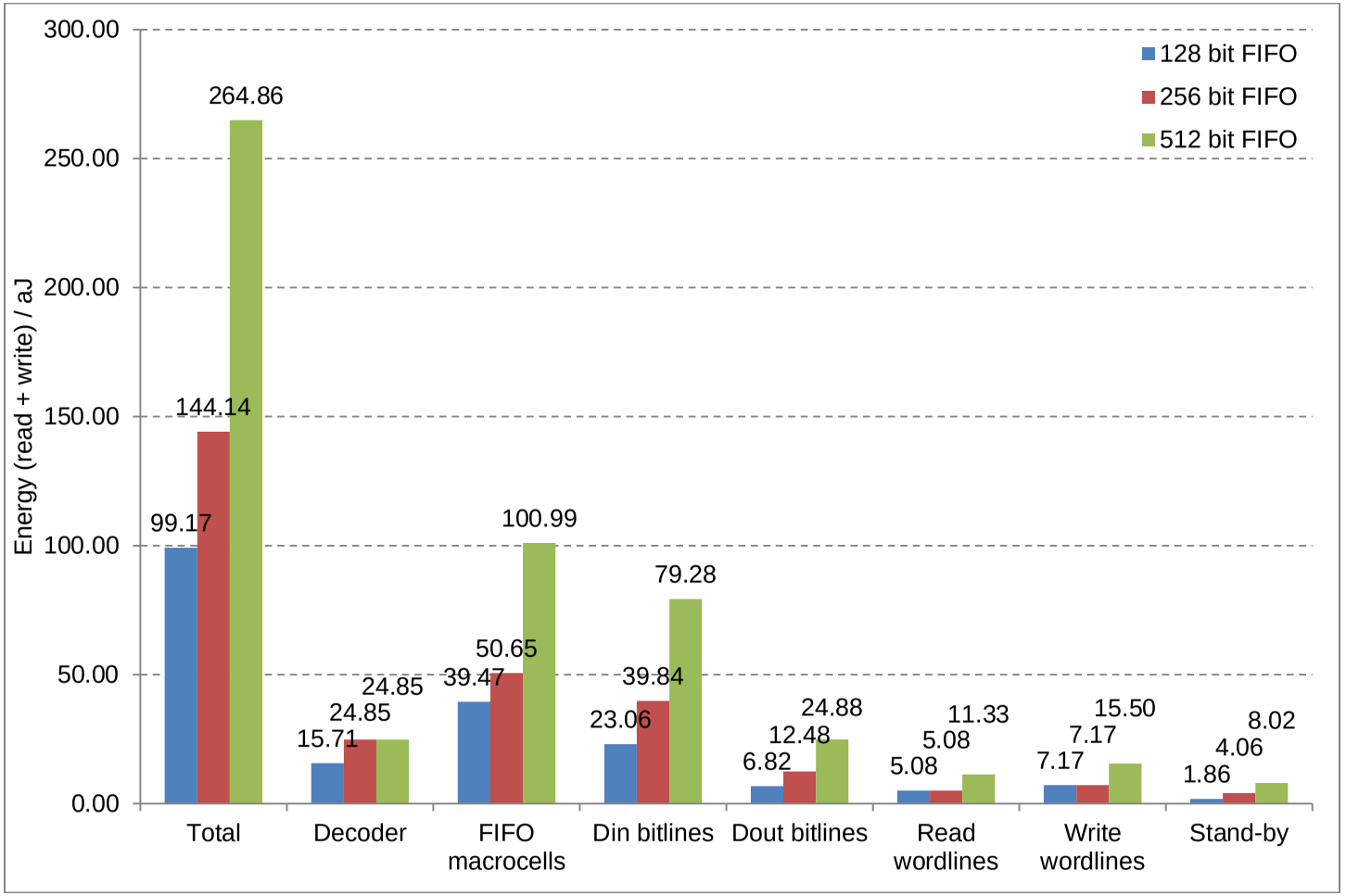


Figure 7.6: FIFO energy per read + write operation.

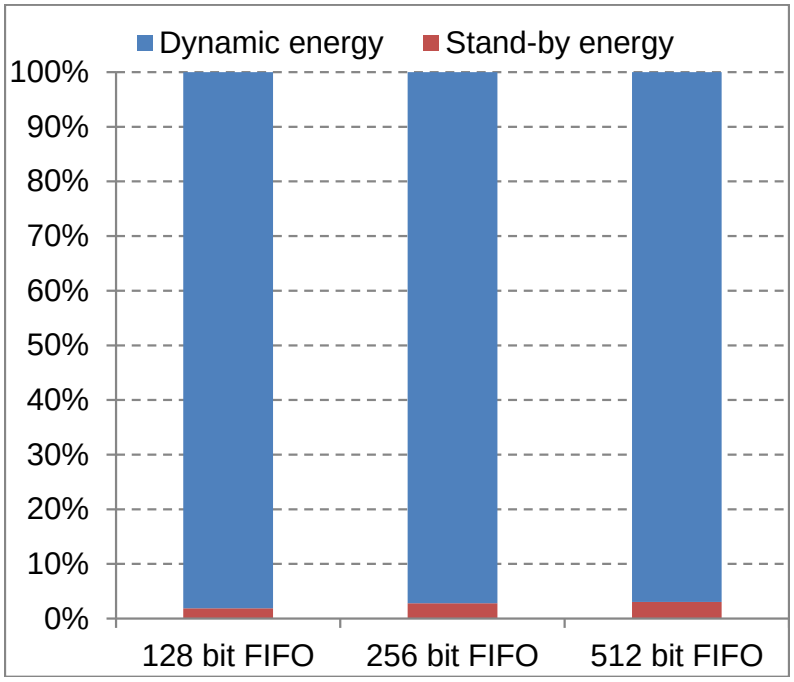


Figure 7.7: FIFO dynamic and stand-by energy breakdown.

Table 7.2: FIFO stand-by energy.

Test case	**128 bit FIFO**		**256 bit FIFO**		**512 bit FIFO**	
	Stand-by energy (aJ)	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**	**Stand-by energy (aJ)**	**% of stand-by JJs**
Read + write	1.86	67.62%	4.06	75.52%	8.02	76.79%
No op	2.76	100.00%	5.38	100.00%	10.45	100.00%

Chapter 8

Summary

Outline

8.1 Completed Work and Discussion

In order to understand the strengths and weaknesses of RQL NDRO-based storage, we designed 13 storage units having different capacities and 32-/64-bit wide pipelines operating at the frequency of 8.5 GHz. We analyzed the effect of doubling the number of bits per row (word) and doubling the number of rows (words) to see scaling trends.

When operating at 8.5 GHz, the units have a low read latency of 1-3 cycles. We believe that with some changes done to the storage unit designs, their clock frequency can be increased to 12-13 GHz if the cycle-wise increase in access latency can be tolerated. The trade-off between the operating frequency and access time of the pipelined on-chip storage and its impact on performance need to be analyzed in the context of complete RQL processor design and application characteristics.

The 4 Kbit register file has the largest complexity (in terms of Josephson junctions) compared to others. The complexity of the 4 Kbit register file is ~1.5X higher than the 4 Kbit RAM. This is the cost of adding the second read port to double read bandwidth in the NDRO-based storage.

The average energy per operation is in the range of ~50-400 aJ at 4.2 K. The highest energy consumption in terms of energy/operation/bit (~9.5 aJ at 4.2 K) is for a write hit in a 2 Kbit 32-bit wide write-back cache. Figure 8.1 on page 97 shows the dynamic and standby energy consumption (en-

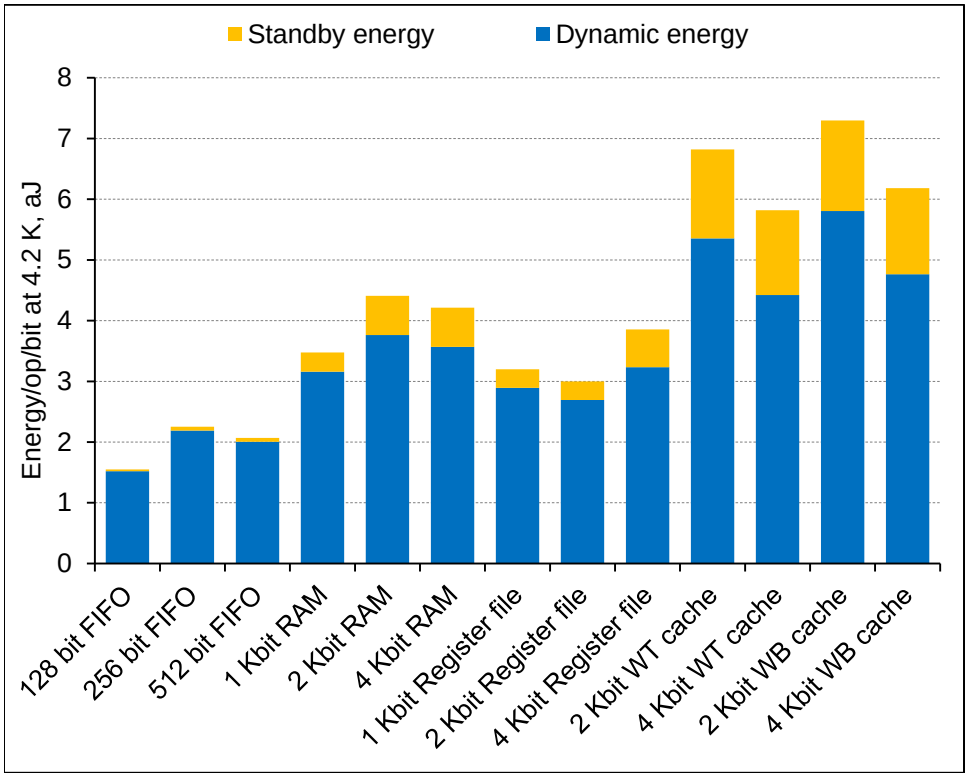


Figure 8.1: The RQL storage energy consumption profile for the 248 nm 100 μA/μm^2 process with the min. I_c = 38 μA at 4.2 K [30].

ergy/operation/bit) of all the storage units. The major contribution comes from the dynamic energy, while the effect of the standby energy becomes noticeable with the increase of the number of inactive cells in multi-Kbit storage designs.

Figure 8.2 on page 98 shows the relative energy and implementation costs of several 32-bit RQL processing units [29] compared to the 1 Kbit 32x32-bit register file (RF) which requires ~ 307 aJ at 4.2 K to write one and read two 32-bit operands. The energy costs of data transfer (e,g,, between on-chip storage and processing units) over superconducting Nb wires are negligible [29].

As shown in Figure 8.2 on page 98, processing by the 32-bit RQL floating-point units takes ~2.5-3.7X more energy than accessing their operands from the RQL register file. The situation is different for integer computing where all 32-bit RQL integer processing units except for the multiplier consume less energy than the 32x32-bit register file.

To calculate the total energy consumption of superconductor circuits, we need to take into account the low efficiency of cryocoolers, which is currently in the range from 0.1 to 0.36% (or in other words, from 360 to 1000 W/W).

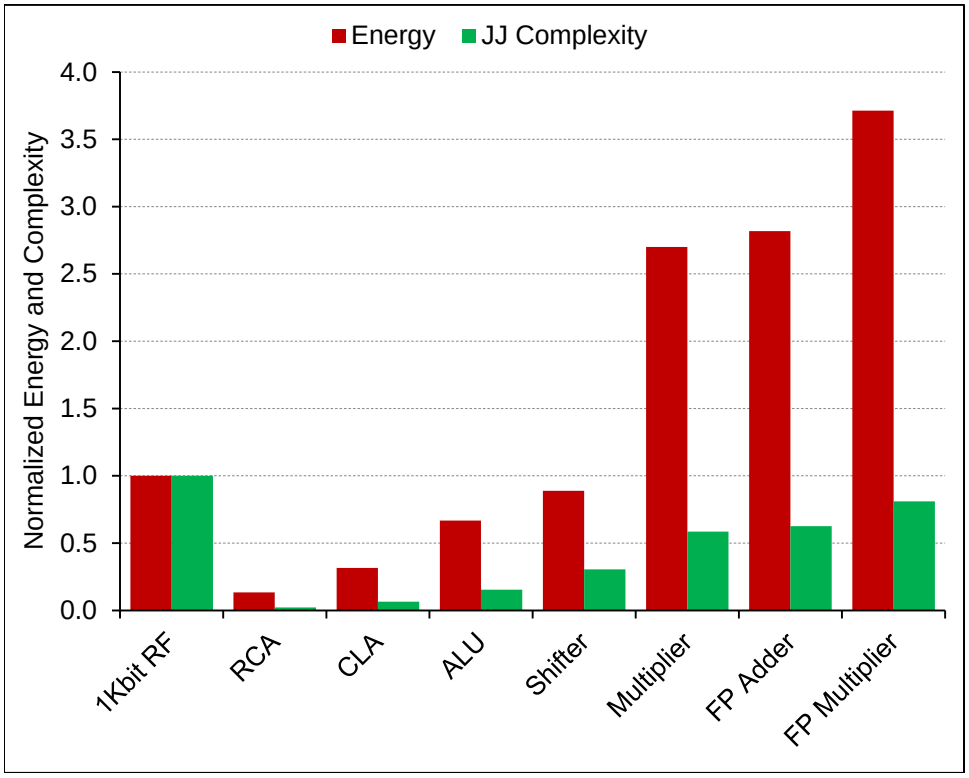


Figure 8.2: Relative energy and JJ complexity of the 32-bit RQL processing units compared to the 8.5 GHz 1 Kbit 32x32 bit register file. The RQL processing units are: a 2.06 GHz non-pipelined ripple-carry adder (RCA), a 12.1 GHz sparse-tree parallel-prefix carry look-ahead adder (CLA), a 16.3 GHz arithmetic-logic unit (ALU), an 13.0 GHz array shifter, an 13.0 GHz integer multiplier, a 12.1 GHz single-precision floating-point (FP) adder, and a 12.1 GHz single-precision floating-point (FP) multiplier [30].

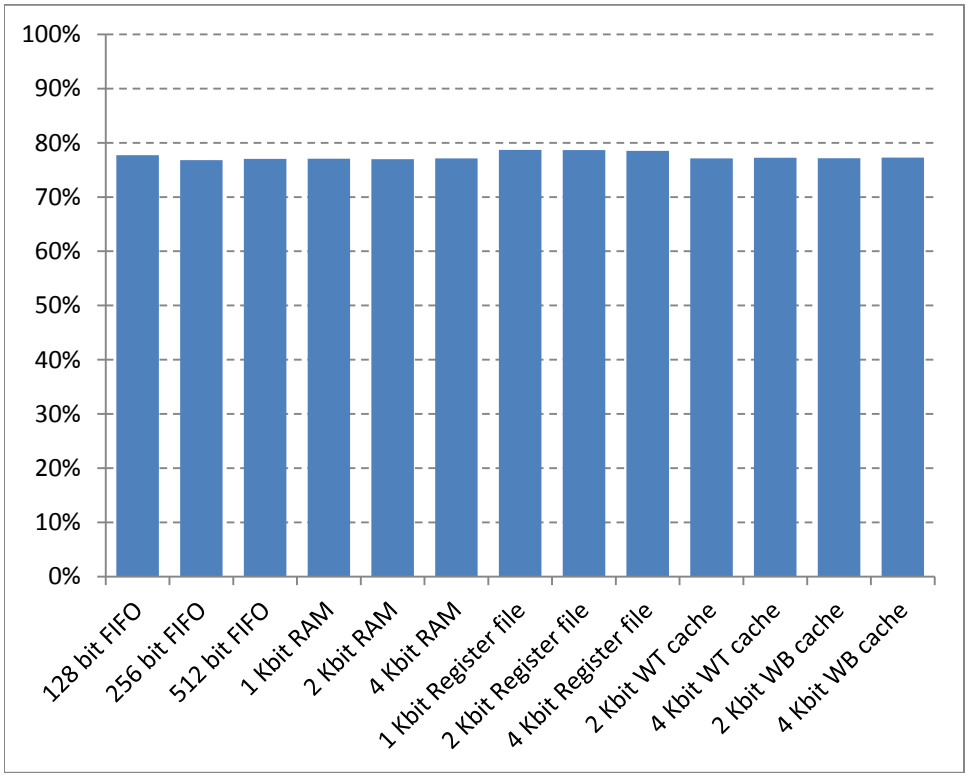


Figure 8.3: The presentage of the connection cells to the total complexity in terms of JJs.

Using the conservative estimate of 1000 W/W, we estimate that our on-chip RQL storage units will consume from 1.5 to 7.2 fJ/operation/bit at room temperature.

Finally, we see the major weakness of the RQL NDRO-based storage, namely its high implementation cost in terms of the number of JJs, which leads to large area and low storage density. The dominant contribution to the JJ complexity comes from the connection cells used for local communication, fan-out increase, delay lines, drivers and receivers for Nb passive transmission lines, as shown in Figure 8.3 on page 99.

We conclude that the RQL NDRO-based storage can provide a high performance energy-efficient solution for on-chip storage units such as FIFOs, local memory, register files, and L1 caches. However, their capacity will be limited due to the high JJ count and large area required for their implementation. The significant improvement in on-chip RQL storage density due to an order of magnitude decrease in the area of RQL cells is expected to happen with the arrival of a new generation of the MIT LL superconductor fabrication process with the minimum feature size below 200 nm.

8.2 Future Work

As we mentioned in the previous section, the frequency of the designs can be increased to 12-13 GHz. This allows our designs to operate together with processing units at higher frequencies. To achieve this, the circuits that in the critical path should be redesigned, the timing of the design may be changed as well.

New generations of the superconductor fabrication process are currently under development at the MIT Lincoln Laboratory. The critical current density of the new process is expected to be 200 $\mu A/\mu m^2$ or even 500 $\mu A/\mu m^2$ [64]. At the same time, the linewidth and via size are expected to decrease. These features will certainly increase the speed of the SFQ circuits by reducing the SFQ gate delays.

The ultimate goal of the superconductor community is to build a complete computing system with superconductor technology. This research helps to understand the challenges and trade-offs in designing energy-efficient on-chip storage for superconductor RQL processors.

Bibliography

[1] K. Likharev and V. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *Applied Superconductivity, IEEE Transactions on*, vol. 1, no. 1, pp. 3 –28, Mar. 1991. (Cited on pages 2 and 7).

[2] B. D. Josephson, "The discovery of tunnelling supercurrents," *Rev. Mod. Phys.*, vol. 46, pp. 251–254, Apr 1974. [Online]. Available: http://link.aps.org/doi/10.1103/RevModPhys.46.251 (Cited on page 2).

[3] B. Josephson, "Possible new effects in superconductive tunnelling," *Physics Letters*, vol. 1, no. 7, pp. 251 – 253, 1962. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031916362913690 (Cited on page 2).

[4] D. K. Brock, "RSFQ technology: Circuits and systems," *Int. J. High Speed ELectron. Syst.,*, vol. 11, pp. 307–362, 2001. (Cited on page 2).

[5] W. Chen, A. Rylyakov, V. Patel, J. Lukens, and K. Likharev, "Rapid single flux quantum T-flip flop operating up to 770 GHz," *Applied Superconductivity, IEEE Transactions on*, vol. 9, no. 2, pp. 3212 –3215, Jun. 1999. (Cited on page 2).

[6] D. S. Holmes, A. L. Ripple, and M. A. Manheimer, "Energy-efficient superconducting computing - power budgets and requirements," *Applied Superconductivity, IEEE Transactions on*, vol. 23, no. 3, pp. 1 701 610–1 701 610, Jun. 2013. (Cited on page 2).

[7] Z. Bao, M. Bhushan, S. Ran, and J. Lukens, "Fabrication of high quality, deep-submicron Nb/AlOx/Nb Josephson junctions using chemical mechanical polishing," *Applied Superconductivity, IEEE Transactions on*, vol. 5, no. 2, pp. 2731 – 2734, Jun. 1995. (Cited on page 2).

[8] L. Lee, E. Arambula, G. Hanaya, C. Dang, R. Sandell, and H. Chan, "RHEA (resist-hardened etch and anodization) process for fine-geometry

Josephson junction fabrication," *Magnetics, IEEE Transactions on*, vol. 27, no. 2, pp. 3133 –3136, Mar. 1991. (Cited on page 2).

[9] Q. Zhong, W. Cao, J. Li, Y. Zhong, and X. Wang, "Study of dry etching process using SF6 and CF4/O2 for Nb/NbxSi1 /Nb Josephson-junction fabrication," in *Precision Electromagnetic Measurements (CPEM), 2012 Conference on*, Jul. 2012, pp. 46 –47. (Cited on page 2).

[10] P. Bunyk, K. Likharev, and D. Zinoviev, "RSFQ technology: Physics and devices," *Int. J. High Speed ELectron. Syst.,*, vol. 11, pp. 257–305, 2001. (Cited on pages 2 and 3).

[11] (2005, Aug.) Superconducting Technology Assessment (STA). National Security Agency. [Online]. Available: http://www.nitrd.gov/pubs/nsa/sta.pdf (Cited on pages 3, 4, 7, 9, and 11).

[12] Y. Akahori and K. Hohkawa, "A Josephson dual-phase AC-powered logic network using special latch circuits," *Electron Devices, IEEE Transactions on*, vol. 32, no. 11, pp. 2339 – 2344, Nov. 1985. (Cited on page 3).

[13] P. Arnett and D. Herrell, "Regulated AC power for Josephson interferometer latching logic circuits," *Magnetics, IEEE Transactions on*, vol. 15, no. 1, pp. 554 – 557, Jan. 1979. (Cited on page 3).

[14] S. Hasuo, "High-speed digital circuits for a Josephson computer," in *Multiple-Valued Logic, 1992. Proceedings., Twenty-Second International Symposium on*, May 1992, pp. 2 –8. (Cited on pages 3, 10, and 11).

[15] V. Michal, E. Baggetta, M. Aurino, S. Bouat, and J. Villegier, "Superconducting RSFQ logic: Towards 100GHz digital electronics," in *Radioelektronika (RADIOELEKTRONIKA), 2011 21st International Conference*, Apr. 2011, pp. 1 –8. (Cited on pages 3 and 7).

[16] Massachusetts Institute of Technology Lincoln Laboratory. Forecasting superconductive electronics technology. National Security Agency. [Online]. Available: https://www.nsa.gov/research/tnw/tnw203/articles/pdfs/TNW203_article2.pdf (Cited on pages 4 and 5).

[17] K. Takagi, M. Tanaka, S. Iwasaki, R. Kasagi, I. Kataeva, S. Nagasawa, T. Satoh, H. Akaike, and A. Fujimaki, "SFQ propagation properties in passive transmission lines based on a 10-Nb-layer structure," *Applied Superconductivity, IEEE Transactions on*, vol. 19, no. 3, pp. 617 –620, Jun 2009. (Cited on pages 5 and 6).

[18] H. Suzuki, S. Nagasawa, K. Miyahara, and Y. Enomoto, "Characteristics of driver and receiver circuits with a passive transmission line in RSFQ circuits," *Applied Superconductivity, IEEE Transactions on*, vol. 10, no. 3, pp. 1637 –1641, Sep 2000. (Cited on page 5).

[19] D. Gupta, W. Li, S. Kaplan, and I. Vernik, "High-speed interchip data transmission technology for superconducting multi-chip modules," *Applied Superconductivity, IEEE Transactions on*, vol. 11, no. 1, pp. 731 –734, Mar 2001. (Cited on page 5).

[20] C. L. Ayala, "Energy-efficient wide datapath integer arithmetic logic units using superconductor logic," Ph.D. dissertation, Stony Brook University, 2012. (Cited on pages 5, 7, and 8).

[21] S. Yorozu, Y. Kameda, H. Terai, A. Fujimaki, T. Yamada, and S. Tahara, "A single flux quantum standard logic cell library," *Physica C: Superconductivity*, vol. 378-381, Part 2, no. 0, pp. 1471 – 1474, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921453402017598 (Cited on page 6).

[22] O. Mukhanov, "Energy-efficient single flux quantum technology," *Applied Superconductivity, IEEE Transactions on*, vol. 21, no. 3, pp. 760 –769, Jun. 2011. (Cited on pages 5 and 8).

[23] D. Kirichenko, S. Sarwana, and A. Kirichenko, "Zero static power dissipation biasing of RSFQ circuits," *Applied Superconductivity, IEEE Transactions on*, vol. 21, no. 3, pp. 776 –779, Jun. 2011. (Cited on page 8).

[24] Y. Yamanashi, T. Nishigai, and N. Yoshikawa, "Study of LR-loading technique for low-power single flux quantum circuits," *Applied Superconductivity, IEEE Transactions on*, vol. 17, no. 2, pp. 150–153, June 2007. (Cited on page 9).

[25] Q. P. Herr, A. Y. Herr, O. T. Oberg, and A. G. Ioannidis, "Ultra-low-power superconductor logic," *arXiv.org*, vol. 21240, p. 7, Mar. 2011. [Online]. Available: http://arxiv.org/abs/1103.4269 (Cited on pages 9, 15, and 16).

[26] O. Oberg, Q. Herr, A. Ioannidis, and A. Herr, "Integrated power divider for superconducting digital circuits," *Applied Superconductivity, IEEE Transactions on*, vol. 21, no. 3, pp. 571 –574, Jun. 2011. (Cited on pages 9 and 15).

[27] O. T. Oberg, "Superconducting logic circuit operating with reciprocal magnetic flux quanta," Ph.D. dissertation, University of Maryland, College Park, 2011. (Cited on pages 9, 15, 16, 18, 19, 20, 21, 22, 23, and 24).

[28] A. Y. Herr, Q. P. Herr, O. T. Oberg, and O. Naaman, "An 8-bit carry look-ahead adder with 150 ps latency and sub-microwatt power dissipation at 10 GHz," *Journal of Applied Physics*, vol. 113, no. 3, pp. 033 911 – 033 911–6, Jan 2013. (Cited on pages 9, 15, 24, and 25).

[29] M. Dorojevets, Z. Chen, C. L. Ayala, and A. K. Kasperek, "Towards 32-bit energy-efficient superconductor RQL processors: The cell-level design and analysis of key processing and on-chip storage units," *Applied Superconductivity, IEEE Transactions on*, vol. 25, no. 3, pp. 1–8, Jun. 2015. (Cited on pages 9, 15, 16, and 97).

[30] M. Dorojevets and Z. Chen, "Fast pipelined storage for high-performance energy-efficient computing with superconductor technology," in *Emerging Technologies for a Smarter World (CEWIT), 2015 12th International Conference Expo on*, Oct 2015, pp. 1–6. (Cited on pages 9, 97, and 98).

[31] M. Dorojevets, *RQL Cell Library Specification*, Ultra-High-Speed Computing Lab, Stony Brook University, NY, Mar. 2012. (Cited on pages 9 and 15).

[32] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Superconductor Science and Technology*, vol. 26, no. 3, p. 035010, 2013. [Online]. Available: http://stacks.iop.org/0953-2048/26/i=3/a=035010 (Cited on page 9).

[33] N. Yoshikawa and Y. Kato, "Reduction of power consumption of RSFQ circuits by inductance-load biasing," *Superconductor Science and Technology*, vol. 12, no. 11, p. 918, 1999. [Online]. Available: http://stacks.iop.org/0953-2048/12/i=11/a=367 (Cited on page 9).

[34] N. Shimizu, Y. Harada, N. Miyamoto, and E. Goto, "A new A/D converter with quantum flux parametron," *Magnetics, IEEE Transactions on*, vol. 25, no. 2, pp. 865–868, Mar 1989. (Cited on page 9).

[35] M. Hosoya, W. Hioe, J. Casas, R. Kamikawai, Y. Harada, Y. Wada, H. Nakane, R. Suda, and E. Goto, "Quantum flux parametron: A single quantum flux device for josephson supercomputer," *Applied Superconductivity, IEEE Transactions on*, vol. 1, no. 2, pp. 77–89, June 1991. (Cited on page 9).

[36] M. Hosoya, W. Hioe, K. Takagi, and E. Goto, "Operation of a 1-bit quantum flux parametron shift register (latch) by 4-phase 36-ghz clock," *Applied Superconductivity, IEEE Transactions on*, vol. 5, no. 2, pp. 2831–2834, June 1995. (Cited on page 9).

[37] M. Tanaka, H. Akaike, A. Fujimaki, Y. Yamanashi, N. Yoshikawa, S. Nagasawa, K. Takagi, and N. Takagi, "100-GHz single-flux-quantum bit-serial adder based on 10- niobium process," *Applied Superconductivity, IEEE Transactions on*, vol. 21, no. 3, pp. 792 –796, Jun. 2011. (Cited on page 10).

[38] S. Kotani, A. Inoue, T. Imamura, and S. Hasuo, "An 8-b Josephson digital signal processor," *Solid-State Circuits, IEEE Journal of*, vol. 25, no. 6, pp. 1518 –1525, Dec. 1990. (Cited on page 10).

[39] S. Kotani, N. Fujimaki, T. Imamura, and S. Hasuo, "A subnanosecond Josephson 16-bit ALU," *Solid-State Circuits, IEEE Journal of*, vol. 23, no. 2, pp. 591 –596, Apr. 1988. (Cited on page 10).

[40] M. Dorojevets and P. Bunyk, "Architectural and implementation challenges in designing high-performance RSFQ processors: A FLUX-1 microprocessor and beyond," *Applied Superconductivity, IEEE Transactions on*, vol. 13, no. 2, pp. 446 – 449, Jun. 2003. (Cited on pages 11 and 12).

[41] P. Bunyk, M. Leung, J. Spargo, and M. Dorojevets, "Flux-1 RSFQ microprocessor: Physical design and test results," *Applied Superconductivity, IEEE Transactions on*, vol. 13, no. 2, pp. 433 – 436, Jun. 2003. (Cited on page 11).

[42] M. Dorojevets, "Architecture and design of an 8-bit FLUX-1 superconductor RFSQ microprocessor," *International Journal of High Speed Electronics and Systems*, vol. 12, no. 02, pp. 521–529, 2002. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0129156402001435 (Cited on page 11).

[43] M. Tanaka, F. Matsuzaki, T. Kondo, N. Nakajima, Y. Yamanashi, A. Fujimaki, H. Hayakawa, N. Yoshikawa, H. Terai, and S. Yorozu, "A single-flux-quantum logic prototype microprocessor," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, Feb. 2004, pp. 298 – 529 Vol.1. (Cited on page 13).

[44] Y. Yamanashi, M. Tanaka, A. Akimoto, H. Park, Y. Kamiya, N. Irie, N. Yoshikawa, A. Fujimaki, H. Terai, and Y. Hashimoto, "Design and

implementation of a pipelined bit-serial SFQ microprocessor, CORE1B," *Applied Superconductivity, IEEE Transactions on*, vol. 17, no. 2, pp. 474 –477, Jun. 2007. (Cited on page 13).

[45] A. Fujimaki, M. Tanaka, T. Yamada, Y. Yamanashi, H. Park, and N. Yoshikawa, "Bit-serial single flux quantum microprocessor core," *IEICE Transactions on Electronics*, vol. E91-C, pp. 342–349, Mar. 2008. (Cited on page 13).

[46] M. Tanaka, Y. Yamanashi, N. Irie, H.-J. Park, S. Iwasaki, K. Takagi, K. Taketomi, A. Fujimaki, N. Yoshikawa, H. Terai, and S. Yorozu, "Design and implementation of a pipelined 8 bit-serial single-flux-quantum microprocessor with cache memories," *Superconductor Science and Technology*, vol. 20, no. 11, p. S305, 2007. [Online]. Available: http://stacks.iop.org/0953-2048/20/i=11/a=S01 (Cited on pages 13 and 14).

[47] M. Dorojevets, C. Ayala, and A. Kasperek, "Data-flow microarchitecture for wide datapath RSFQ processors: Design study," *Applied Superconductivity, IEEE Transactions on*, vol. 21, no. 3, pp. 787 –791, Jun. 2011. (Cited on page 14).

[48] T. Filippov, M. Dorojevets, A. Sahu, A. Kirichenko, C. Ayala, and O. Mukhanov, "8-bit asynchronous wave-pipelined RSFQ arithmetic-logic unit," *Applied Superconductivity, IEEE Transactions on*, vol. 21, no. 3, pp. 847 –851, Jun. 2011. (Cited on pages 14 and 26).

[49] T. V. Filippov, A. Sahu, A. F. Kirichenko, I. V. Vernik, M. Dorojevets, C. L. Ayala, and O. A. Mukhanov, "20 GHz operation of an asynchronous wave-pipelined RSFQ arithmetic-logic unit," *Physics Procedia*, vol. 36, no. 0, pp. 59 – 65, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1875389212020676 (Cited on pages 14 and 26).

[50] A. F. Kirichenko, T. V. Filippov, A. Sahu, O. A. Mukhanov, M. Dorojevets, and A. K. Kasperek, "Demonstration of RSFQ 8-bit multi-port register file." in *Proceedings of Applied Superconductivity Conference 2012 (ASC '12)*, Portland, OR., Oct. 2012. (Cited on page 14).

[51] A. Kirichenko, A. Sahu, T. Filippov, O. Mukhanov, A. Dotsenko, M. Dorojevets, and A. Kasperek, "Demonstration of an 8x8-bit RSFQ multi-port register file," in *Superconductive Electronics Conference (ISEC), 2013 IEEE 14th International*, Cambridge, MA, July 7-11 2013, pp. 1–3. (Cited on page 26).

[52] S. K. Tolpygo, V. Bolkhovsky, T. Weir, W. D. Oliver, L. M. Johnson, and M. Gouker, "MIT LL superconductor electronics fabrication process for VLSI circuits with 4, 8 and 10 niobium layers," in *IEEE/CSC Superconductivity News Forum (Global Edition)*, Oct. 2014. (Cited on pages 28 and 29).

[53] M. Manheimer, "Cryogenic computing complexity program: Phase 1 introduction," *Applied Superconductivity, IEEE Transactions on*, vol. 25, no. 3, pp. 1–4, June 2015. (Cited on page 29).

[54] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, A. Wynn, D. E. Oates, L. M. Johnson, and M. A. Gouker, "Advanced fabrication processes for superconducting very large scale integrated circuits," *ArXiv e-prints*, Sep. 2015. (Cited on pages 29 and 30).

[55] S. Nagasawa, H. Hasegawa, T. Hashimoto, H. Suzuki, K. Miyahara, and Y. Enomoto, "Superconducting latching/SFQ hybrid RAM," *Applied Superconductivity, IEEE Transactions on*, vol. 11, no. 1, pp. 533 –536, Mar. 2001. (Cited on pages 31 and 32).

[56] S. Nagasawa, K. Hinode, T. Satoh, Y. Kitagawa, and M. Hidaka, "Design of all-DC-powered high-speed single flux quantum random access memory based on a pipeline structure for memory cell arrays," *Superconductor Science and Technology*, vol. 19, no. 5, p. S325, 2006. [Online]. Available: http://stacks.iop.org/0953-2048/19/i=5/a=S34 (Cited on pages 31 and 32).

[57] M. Hidaka, S. Nagasawa, K. Hinode, and T. Satoh, "Improvements in fabrication process for Nb-based single flux quantum circuits in Japan," *IEICE Transactions on Electronics*, vol. 91, pp. 318–324, 2010. (Cited on page 31).

[58] T. V. Duzer, L. Zheng, S. R. Whiteley, H. Kim, J. Kim, X. Meng, and T. Ortlepp, "64-kb hybrid Josephson-CMOS 4 Kelvin RAM with 400 ps access time and 12 mW read power," *Applied Superconductivity, IEEE Transactions on*, vol. 23, no. 3, pp. 1 700 504,1 700 504, Jun. 2013. (Cited on pages 31, 32, and 33).

[59] I. Vernik, V. Bol'ginov, S. Bakurskiy, A. Golubov, M. Kupriyanov, V. Ryazanov, and O. Mukhanov, "Magnetic josephson junctions with superconducting interlayer for cryogenic memory," *Applied Superconductivity, IEEE Transactions on*, vol. 23, no. 3, pp. 1 701 208–1 701 208, June 2013. (Cited on pages 31, 34, and 71).

[60] B. Baek, S. Benz, W. Rippard, S. Russek, P. Dresselhaus, M. Pufall, and H. Rogalla, "Magnetic barrier structures for superconductingmagnetic hybrid josephson junctions," in *Superconductive Electronics Conference (ISEC), 2013 IEEE 14th International*, July 2013, pp. 1–3. (Cited on pages 31, 34, and 71).

[61] A. Herr and Q. Herr, "Josephson magnetic random access memory system and method," Sep. 18 2012, uS Patent 8,270,209. [Online]. Available: https://www.google.com/patents/US8270209 (Cited on pages 31, 34, and 71).

[62] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 2012. (Cited on pages 34 and 75).

[63] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd ed. Pearson, 2005. (Cited on pages 37, 39, and 75).

[64] S. Tolpygo, V. Bolkhovsky, T. Weir, L. Johnson, M. Gouker, and W. Oliver, "Fabrication process and properties of fully-planarized deep-submicron Nb/Al-AlOx/Nb josephson junctions for vlsi circuits," *Applied Superconductivity, IEEE Transactions on*, vol. 25, no. 3, pp. 1–12, June 2015. (Cited on page 100).