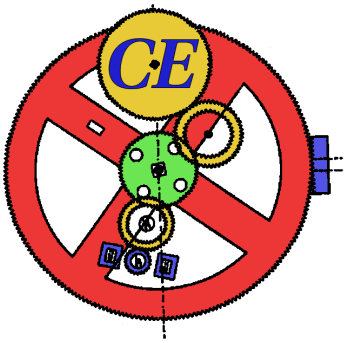


Computer Engineering
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ce.et.tudelft.nl/

2005

MSc THESIS

Arithmetic Operations in Single Electron Tunneling Technology

CE-MS-2005-01

Abstract

In this thesis we investigate the implementation of arithmetic operations in Single Electron Tunneling (SET) technology. In particular we focus on design methodologies for SET based circuits, high radix addition in the Electron Counting (EC) paradigm, and the computation of periodic symmetric functions. First, we propose a design methodology that, given a circuit topology and the corresponding targeted behaviour, assists the circuit designer in deriving the circuit parameters in an analytical way. The methodology, based on the mathematical description of the tunnel junctions in the circuit, allows for a time effective design of SET based building blocks. Moreover the method allows for the adaptation of such blocks for the utilization in larger SET circuits. The methodology we propose can be used for circuits operating under the Single Electron Encoded Logic (SEEL) paradigm and it is very useful as the design of SET circuits is mostly manually done because there are no tools and only few simulators are available. Second, we investigate the implementation of high radix addition in the EC paradigm, which is based on encoding integer values as a net charge. In previous research a high radix addition scheme was proposed, that assumes the presence of a conditional charge movement ($MCke$) block, which was only described as a black box. We propose five possible $MCke$ block implementations, each of which is described in detail. One of the proposed $MCke$ implementations is utilized in the design of a 6-bit radix-8 adder. Finally, we investigate the implementation of Periodic Symmetric Functions (PSFs) in SET technology. A building block is proposed that performs a multiple input PSF. The block we propose can be used for the computation of any function that is or can be expressed as a PSF, thus it can be utilized for the implementation of a large number of arithmetic operations, e.g., addition, parity check, multi-operand addition, as they belong to the class of generalized PSFs. Two example circuits, a 3-bit adder and a Block Save Adder (BSA(4,2,4)) are presented and verified by means of simulation. All designs included in this thesis are verified by means of simulation.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Arithmetic Operations in Single Electron Tunneling Technology

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

C.H. Meenderinck
born in Amsterdam, The Netherlands

Computer Engineering
Department of Electrical Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Arithmetic Operations in Single Electron Tunneling Technology

by C.H. Meenderinck

Abstract

In this thesis we investigate the implementation of arithmetic operations in Single Electron Tunneling (SET) technology. In particular we focus on design methodologies for SET based circuits, high radix addition in the Electron Counting (EC) paradigm, and the computation of periodic symmetric functions. First, we propose a design methodology that, given a circuit topology and the corresponding targeted behaviour, assists the circuit designer in deriving the circuit parameters in an analytical way. The methodology, based on the mathematical description of the tunnel junctions in the circuit, allows for a time effective design of SET based building blocks. Moreover the method allows for the adaptation of such blocks for the utilization in larger SET circuits. The methodology we propose can be used for circuits operating under the Single Electron Encoded Logic (SEEL) paradigm and it is very useful as the design of SET circuits is mostly manually done because there are no tools and only few simulators are available. Second, we investigate the implementation of high radix addition in the EC paradigm, which is based on encoding integer values as a net charge. In previous research a high radix addition scheme was proposed, that assumes the presence of a conditional charge movement ($MCke$) block, which was only described as a black box. We propose five possible $MCke$ block implementations, each of which is described in detail. One of the proposed $MCke$ implementations is utilized in the design of a 6-bit radix-8 adder. Finally, we investigate the implementation of Periodic Symmetric Functions (PSFs) in SET technology. A building block is proposed that performs a multiple input PSF. The block we propose can be used for the computation of any function that is or can be expressed as a PSF, thus it can be utilized for the implementation of a large number of arithmetic operations, e.g., addition, parity check, multi-operand addition, as they belong to the class of generalized PSFs. Two example circuits, a 3-bit adder and a Block Save Adder (BSA(4,2,4)) are presented and verified by means of simulation. All designs included in this thesis are verified by means of simulation.

Laboratory : Computer Engineering
Codenumber : CE-MS-2005-01

Committee Members :

Advisor:	Sorin Coţofană, CE, TU Delft
Chairperson:	Stamatis Vassiliadis, CE, TU Delft
Member:	Jaap Hoekstra, ELCA, TU Delft
Member:	Koen Bertels, CE, TU Delft

Contents

List of Figures

Figures denoted by a † were (partly in some cases) drawn by C.R. Lageweg and used with permission.

List of Tables

Acknowledgements

Nine years have passed since I first started my study at the Faculty of Electronic Engineering at Delft University of Technology. Most of those years were filled with illness and inability to study. I am grateful to have reached the point at which I can present my master's thesis.
First of all I would like to thank my parents for raising me and for their support in everything I did. Especially their compassion and great help during the time that I was coping with my illness, deserves great honor.
I would like to thank all who helped me during the time of my illness. Many thanks to those who helped me with the practical things that I was unable to do. Also many thanks to those who prayed for me. Thanks to my church and cell group who stood beside me.
I would like to thank professor Stamatis Vassiliadis for his warm welcome to the Computer Engineering laboratory. Thank you for giving me all opportunity to finish my study, despite the severe health problems.
I am most grateful for my advisor Sorin Coţofană, who, right from the start, showed great confidence in me. Sorin was not only my advisor, he also was my teacher in many different area's concerning the academic world. I hold him in great respect for what he does and who he is. I trust we will have four more good years of cooperation ahead of us.
I also would like to thank Philip Post for the great times of conversation and snooker. Thanks to Walter Harrewijn for his contribution to my presentations.
Above all I would like to thank my heavenly Father, who blessed me abundantly. He gave me the strength to continue, and above all healed me through a miracle. Praise be to Jesus, who gave Himself for me; through His sacrifice I have received true life and by His stripes I was healed.

C.H. Meenderinck
Delft, The Netherlands
June 30, 2005

1 Introduction

Since the introduction of the computer we have seen an ongoing rise in the complexity of software programs. Computer programs are replacing human labor in an increasing way. Software engineers are more and more capable of embedding complex human skills in programs. The desire for efficient labour, which comes from economic reasons, stimulated the design of labor aiding software. This software has been and is growing in complexity also and nowadays it enables us to accomplish things, which could not have been done without computers.

Because of this ongoing increase in complexity of software, there is a ongoing need for more powerful computers. Moore's law (which states that the number of components per chip doubles every 18 months) has remained valid since it first became known in 1965. Though it seems that on the area of paradigm and technology it becomes troublesome to stay in track with the law. It is generally expected that current technology eventually cannot be pushed beyond some limit. This limit is expected to arise in mainly two areas: power consumption and scalability. The International Technology Roadmap for Semiconductors (ITRS) [32] states: "however, since 2001, we have reached the point where the horizon of the Roadmap challenges the most optimistic projections for continued scaling of CMOS." Therefore in the Roadmap post-CMOS devices were included.

One of the alternative technologies for creating these "novel" devices is Single Electron Tunneling (SET) technology. SET technology allows controlled transport of single electrons among quantum islands. If values or bits are represented by single or few electrons this technology has in principal the potential of performing computation with lower power consumption than CMOS technology. When semiconductor structures are scaled down to the nanometer region and beyond, quantum mechanics effects increases and finally determine the behaviour. While for CMOS this causes faults to occur, and therefore is a limit to the scalability, for SET this means scalability even on atomic scale set, because SET is based on quantum mechanics effects [35, 6].

Because of the SET technology potential several proposals have been made to implement computational structures using SET technology. These implementations are mainly categorized in two types [35, 28]. The first type of implementation represents logic value by voltage [33, 26, 8] and SET devices are used to switch currents on and off, resulting in MOSFET like behaviour. The main advantage of this approach is that current designs can easily be translated into SET technology, but there are disadvantages as well. A major drawback to this approach is the large power dissipation [21, 5]. A second drawback is the inefficient use of the device because SET devices are capable of controlling the transport of single electrons.

The second type of implementation represents bits by single electrons, thus it utilize the SET device capability to control the transport of single electrons. In these implementations SET devices are used to transport electrons to and from quantum islands

(sometimes called quantum dots). Using this approach the power consumption is very low. For an overview of these implementations the reader is referred to [28, 35]. Single Electron Encoded Logic (SEEL) is one of these implementations and was proposed by our research group [23, 24, 25]. For other examples of this approach the reader is referred to [3, 20].

These attempts of scientists to build circuits using SET technology have mainly taken place in the last decade, although the principles of SET have been known for many decades. Since SET technology is different from MOS technology, existing design methodologies are not applicable, especially not for implementations of the second type. Additionally, CAD tools are not available and only a small number of simulators exist [1, 2]. Thus the design of SET circuits is generally speaking not automated. Therefore the first objective of this research is to propose a design methodology for deriving the circuit parameters of SET based building blocks, focussed on implementations of the second type. A SET building block is a small circuit consisting of a few (typically between one and four) tunnel junctions and some other circuit elements (like capacitors, voltage sources, etc.), which performs an elementary operation and can be connected to other building blocks to perform more complex operations.

Though developments in the second type of implementation are promising, they do not seem to use the full potential offered by SET technology. Therefore the Electron Counting (EC) paradigm was proposed [9]. The electron counting paradigm is based on representing integer values by number of electrons. Electrons are kept in charge reservoirs and arithmetic computation is done among these reservoirs. Using this paradigm, algorithms and circuits for performing addition/subtraction and multiplication have been devised [10]. Within the EC paradigm context, the addition of two n-bit operands can be computed with a depth-2 network, while the multiplication of two n-bit operands can be computed with a depth-3 network. Both circuits require an $O(n)$ area. To put these results in perspective we mention here that in SEEL fast addition and multiplication can be done using carry-lookahead adders and tree multipliers [22], respectively, which require $O(log\ n)$ time. The SEEL addition scheme requires $O(n\ log\ n)$ area and the multiplier requires even $O(n^2)$ area, therefore both being considerable larger as the EC circuits.

The prospects for EC circuits are good in theory, however practical aspects will limit the effective performance. First of all, there is a limit to the analogue value that can be processed. Generally speaking any EC computation scheme requires building blocks that transport electrons to a charge reservoir and building blocks that sense the electrons in the charge reservoir. Given a certain charge reservoir, both type of building blocks have a limit to the number of electrons they can respectively transport or sense correctly. Thus there is a limit to the operand size of the EC addition/subtraction and multiplication schemes. The second aspect of the effective performance of the EC schemes is delay. For n-bit addition one of the building blocks has to transport 2^{n-1} electrons. Electrons tunnel one after another, so the delay is expected to be proportional to the number of electrons. Thus for large n the delay may become too large to benefit the capabilities electron counting offers.

Because of these two aspects, high radix computation in EC was proposed in [10]. Although a high radix addition scheme was designed and a high radix multiplication

scheme was suggested no complete circuit support was provided. Therefore the second objective of this research is to give circuit support for the existing algorithms in the EC paradigm. Specifically, we focus on the design of a conditional charge transport building block that is a key element in the implementation of EC based high radix addition.

The electron counting paradigm uses specific properties of SET technology, e.g., Coulomb oscillations, to perform operations on amounts of electrons. These properties of SET technology could also be used to perform logic and arithmetic operations on Boolean values. In this line of reasoning, for the last objective of this research, we assume the electron trap, a basic SET structure that exhibits periodic behaviour, and use it as a basis for the implementation of Periodic Symmetric Functions (PSFs). As a large number of arithmetic operations, including addition and parity check, can be expressed as PSFs, such a building block can be used to compute a variety of mathematical operations.

This thesis is organized as follows. Chapter 2 provides a background on the SET phenomenon and introduces the electron counting paradigm in detail, including the building blocks used to perform addition and multiplication in EC. In Chapter 3 the first objective of this research is met by proposing a design methodology for SET building blocks. Chapter 4 concerns the second objective and proposes five implementations of the conditional charge transport block ($MCke$). Using this building block a high radix addition scheme is implemented and simulated. The last objective of this research is met in Chapter 5 in which a scheme is proposed that allows to implement a generalized periodic symmetric function in SET technology. Two examples are provided to illustrate how arithmetic operations, that can be described as generalized PSFs, are implemented using this scheme. Chapter 6 concludes this thesis. Appendix A gives an overview of the basic set topologies used throughout this thesis and provides the most important equations describing these structures. Appendix B contains the source code of a Matlab program used to solve the set of non-linear equations derived in Chapter 5.

2 SET Background

In this chapter some background information on SET technology is provided. In the first section the basic theory of single electron tunneling is presented. Since this thesis is about the utilization of basic SET devices for computation, and not about the physics of SET devices only the theory needed for understanding the SET tunnel junction behaviour is presented. For an in dept presentation of the physics of SET devices the reader is referred to [16, 27, 36]. In Section 2.2 an overview of the principal findings of the research done so far in the area of EC is given.

2.1 Single Electron Tunneling

In the classic physics theory electrons are viewed of as particles and the theory does not allow electrons to cross a barrier like a piece of insulator. In 1923 L. de Broglie [11] suggested that particles may also behave like waves. Three years later this hypothesis was formally described by Schrödinger (see for example [13] for the Schrödinger theory), which became the basis for the quantum mechanics theory of today. Using the Schrödinger wave equation there is a probability that an electron tunnels through a barrier and enters a classically forbidden region.

This tunneling phenomenon is the basic principle of SET technology. It is used to create the basic circuit element of SET technology, the tunnel junction. The symbol for the tunnel junction is depicted in Figure 2.1. The junction is created by separating two conductors with a thin insulator and therefore it behaves like a capacitor. Given that the insulator is thin enough quantum tunneling may occur. The tunneling of an electron through the junction is called a tunnel event.

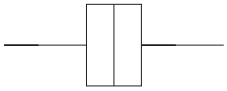
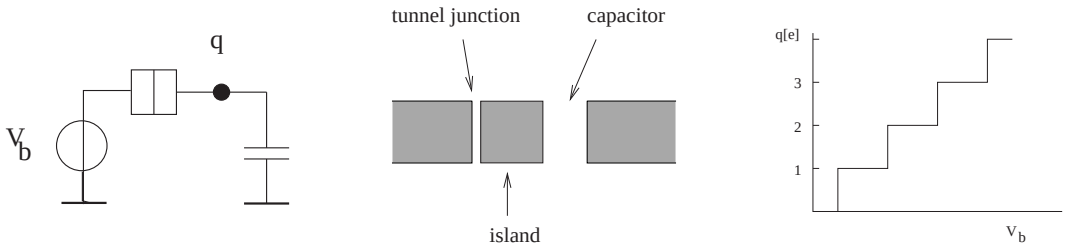


Figure 2.1: Symbol of the tunnel junction.

The electron box (see Figure 2.2) is one of the most simple SET circuits. Node q is a quantum island. On one side it is connected to a voltage source through a tunnel junction, which is formed by the thin piece of insulator. On the other side it is connected to the ground through a capacitor, which is formed by the thick piece of insulator. For an electron to tunnel from the island through the junction, the Coulomb energy $E_C = \frac{q_e^2}{2C}$, where C is the capacitance of the island and q_e is the charge of an electron ($1.60217 * 10^{-19}$ C), is needed. If the Coulomb energy is not available a tunnel event cannot happen. This phenomenon is known as Coulomb blockade.



(a) Circuit. (b) Schematic structure. (c) Transfer function.

Figure 2.2: SET electron box.

The voltage source can provide the energy needed for an electron to tunnel. If the bias voltage exceeds a certain value an electron tunnels from the island. Further increase of the bias voltage eventually leads to a second electron to tunnel. This way, the number of electrons present on the island can be controlled by the bias voltage.

If the voltage source would be the only source of energy, things would have been perfect. Unfortunately, also the thermal energy can provide the energy needed for a tunnel event. Since this is not desired, the thermal energy k_BT should be much smaller than the Coulomb energy needed for a tunnel event. This is expressed as

$$E_C = \frac{q_e^2}{2C} >> k_BT, \tag{2.1}$$

where k_B is the Boltzmann's constant ($1.38066 * 10^{-23} JK^{-1}$) and T is the absolute temperature in Kelvin. For room temperature operation this means that the island capacitance should be in the order of magnitude of $1aF$ or less. Although this condition is valid for most circuits presented in this thesis, simulations were done assuming a $0K$ temperature. Given that the focus of this thesis is on the computational aspects of SET circuitry, by assuming a $0K$ temperature we assured that the simulation results showed functional behaviour only.

There is an other condition for observing the tunneling phenomenon. In classical theories an electron was assumed to be well localized. However, in the quantum mechanics theory electrons are described by wave functions, indicating the probability of the presence of an electron. If a tunnel barrier is insufficiently opaque the electron wave function extends through the barrier and the electron is not clearly localized on the island. The opaqueness of a tunnel barrier is described by the tunnel resistance R_t. A sufficient condition [30, 4] for observing SET charging effects is:

$$R_t >> \frac{h}{q_e^2} = 25.6k\Omega, \tag{2.2}$$

where h is Plank's constant ($6.62607 * 10^{-34} Js$). For further explanation of the tunnel resistance and the derivation of this equation we refer the reader to [36, 15, 37]. In this research we assume a tunnel resistance R_t of $100k\Omega$, which is a commonly used value.

Method of Critical Voltage

When building circuits with tunnel junctions, one needs to predict the tunneling of electrons. This could be done by calculating all free energy in the circuit. But even for modest sized circuits this method results in very complex computation [36]. Therefore generally the method of critical voltage is employed [14, 12, 17]. This method predicts that an electron only tunnels if the voltage across the tunnel junction (V_j) is greater than some critical voltage (V_c).

$$|V_j| > V_c \tag{2.3}$$

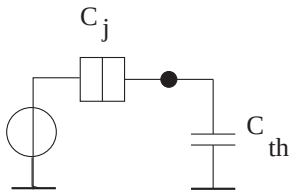


Figure 2.3: Thevenin equivalent circuit for determining the critical voltage.

To calculate the critical voltage of a tunnel junction the *Thevenin equivalent circuit* has to be determined. All but one voltage sources are disregarded while all capacitors and other junctions are reduced to one capacitor (C_{th}). The remaining circuit is the Thevenin equivalent circuit as depicted in Figure 2.3. The critical voltage of the tunnel junction can now be described as

$$V_c = \frac{q_e}{2(C_j + C_{th})}. \tag{2.4}$$

A tunnel junction is called *stable* if the voltage across the junction is smaller than the critical voltage. A circuit is called stable if all junctions are stable. If a junction is unstable, the junction voltage is greater than the critical voltage, an electron tunnels after some time. Quantum tunneling is a probabilistic or stochastic process, which means we can only give the probability that an electron tunnelled. The longer we wait, the bigger the probability of a tunnel event to take place, but there always remains some probability that no electron tunnelled, and thus some error occurred. The error probability $P_{error}(t_d)$ is the probability that, in an unstable circuit, after t_d seconds no tunnel event did take place. This error probability depends on the tunnel rate, which is the average number of tunnel events per second, and can be expressed as follows:

$$\Gamma = \frac{|V_j| - V_c}{q_e R_t} \tag{2.5}$$

Considering a circuit consisting of multiple junctions, assuming they all have the same tunnel rate, the error probability is described as

$$P_{error}(t_d) = e^{-\Gamma t_d} \tag{2.6}$$

in which t_d is the time in seconds from the start of the process. This equation can be rewritten to calculate the required delay time for some error probability.

$$t_d = \frac{-\ln(P_{error})}{\Gamma} \tag{2.7}$$

In this thesis we assume an error probability $P_{error} = 10^{-8}$ and a tunnel resistance $R_t = 10^5\Omega$, which are generally accepted values.

From these equations it can be seen that downscaling a SET circuit is advantageous for its performance. Downscaling results in smaller capacitance values of the quantum islands, tunnel junctions and capacitors, and assuming the charges remain the same, this causes the voltages to increase. As the voltages increase, so does the term $|V_j| - V_c$ of Equation (2.5) increase equally and therefore the tunnel rate increases. According to Equations (2.6) and (2.7) this results in a decrease of the error probability or of the required delay time and hence the performance increases.

Using the method of critical voltage SET circuits can be designed and analyzed. Several basic building blocks were proposed by our research group (see [22] for an overview) of which most perform operations similar to the digital gates used in current CMOS technology. However, the class of building blocks required in the EC paradigm is completely different. In the next section this paradigm is explained and the building blocks, specific for this paradigm and designed prior to the research of this thesis, are presented.

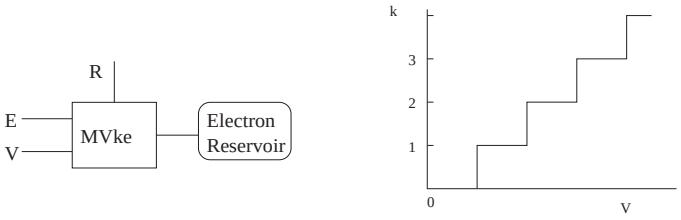
2.2 Electron Counting Paradigm

In the previous section we presented the basics of single electron tunneling. In this section we introduce the Electron Counting paradigm and show how it can be implemented in SET technology. For a complete overview of the research done so far in Electron Counting field, the reader is referred to [22, 10].

Using SET technology single electrons can be controlled and this inspired the novel idea of representing integer numbers by amounts of electrons. Once integer values have been encoded as amounts of electrons, arithmetic operations can be performed directly on electron charges. This is the basic idea behind the EC paradigm. Addition and multiplication based on the EC paradigm are presented in Section 2.2.3 and 2.2.4, respectively. In the next two subsections two building blocks for implementing these operations are discussed. The first building block, the $MVke$ block, can move a certain amount of electrons and it is used to convert a digital integer value into an amount of electrons (analog value). The second building block, the PSF block performs a periodic symmetric function on an analog value and it is used to convert an analog value back into a digital value.

2.2.1 The $MVke$ Block

The first building block is the $MVke$ block which is capable of adding or removing Vk electrons from a charge reservoir, where V is some variable input value and k is a built in constant. The $MVke$ block symbol is depicted in Figure 2.4.



(a) Symbol. (b) Transfer function (E=$'high'$,R=0).

Figure 2.4: $MVke$ building block.

The $MVke$ building block has three inputs and one output, which is the charge reservoir. The first input V controls the transport of electrons and can take any value. The sign of input V determines the direction in which electrons are moved, thus electrons can both be added to or subtracted from the charge reservoir. The magnitude of input V determines the amount of electrons that are moved. If input V is zero, no electrons are transported. The second input E is the enable, which is a logic signal. If E is zero no electrons are moved, no matter what input V is. The last input R is the reset, which is also a logic signal. The reset is used to bring the $MVke$ block back in the initial neutral state.

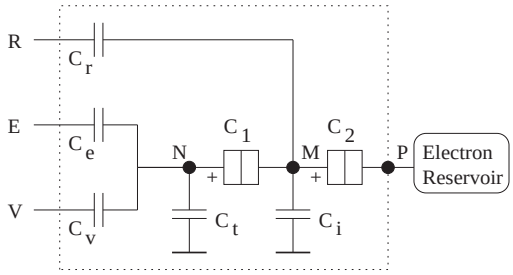
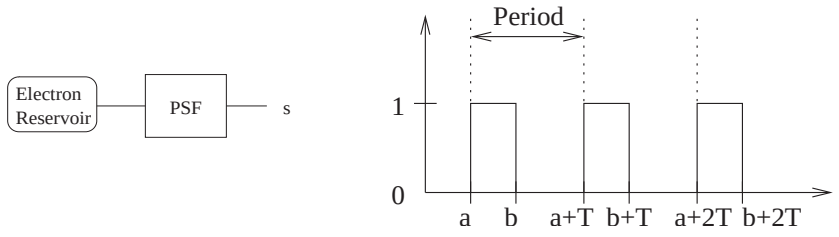


Figure 2.5: $MVke$ building block implementation.

A possible implementation of the $MVke$ building block is depicted in Figure 2.5. The heart of this block are the two tunnel junctions. This structure of two tunnel junctions is often referred to as a SET transistor. The intermediate node is called the gate, while the outer nodes are called the source and drain. For more information on the SET transistor we refer the reader to [26, 16, 35].

The implementation of the $MVke$ block operates as follows. Integers are represented by the absence of electrons in the charge reservoir, which is equal to the presence of a positive charge. While R (reset) and V are zero and input E (enable) is set to $'high'$, the voltage across junction 1 (C1) gets very close to its critical voltage. If at the same time V is set to $'high'$ the critical voltage is exceeded and one electron tunnels from node M to node N. As a result of this event a positive charge is present on node M, which causes the voltage of junction 2 (C2) to exceed the critical voltage of junction 2. So one electron tunnels from node P, which actually is the charge reservoir, to node



(a) Symbol. (b) Transfer function.

Figure 2.6: PSF building block.

M. This process of two tunnel events continues until the voltage across junction 1 has dropped below its critical voltage again. The number of electrons that is removed from the charge reservoir is proportional to the magnitude of both V and C_v. Therefore in this implementation the value of k is determined by, among other things, the value of C_v.

In this implementation electrons can only be removed from a charge reservoir. If the voltages of inputs E, R are made negative this implementation can be used to add electrons to the charge reservoir.

The circuit can be reset to its initial state by setting R to $'high'$ while the other inputs are zero. The positive voltage on R causes a negative voltage across junction 1 which exceeds the critical voltage. One electron tunnels from node N to node M, causing junction 2 into an unstable state, so an electron tunnels to the reservoir. This process continues until all extra electrons present on node N are returned to node P and the charge reservoir is back into the neutral state.

2.2.2 The PSF Block

With the $MVke$ block electrons can be removed from charge reservoirs. Using the PSF block one can determine the number of electrons present in a charge reservoir. The symbol and the transfer function of the PSF block are depicted in Figure 2.6. This building block implements a Periodic Symmetric Function (PSF) $F_p(X)$. A Boolean symmetric function $F_s(x_0, x_1, ..., x_{n-1})$ is a Boolean function for which the output depends on the sum of the inputs $X = \sum_{i=1}^{n-1} x_i$. For a Periodic Symmetric Function it holds true that $F_p(X) = F_p(X + T)$, where T is the period.

The SET electron trap, depicted in Figure 2.7, has a periodic function and can be used as a basis to implement the PSF building block. The SET electron trap functions as follows. If the input voltage rises, the output voltage follows, due to capacitance division. At some point, though, the voltage across the tunnel junction exceeds the critical voltage and an electron tunnels to the output node. As consequence of this tunneling the output voltage drops below the critical voltage. As the input voltage continues to rise, the output voltage rises again until it reaches the critical voltage. The resulting transfer function of the SET electron trap is a periodic function as depicted in Figure 2.7(b).

The relation between the input voltage V_{in} and the output voltage V_{out} of the electron

trap was derived as

$$V_{in} = \frac{C_{\Sigma o}}{C_i} V_{out} + \frac{q_o}{C_i}, \tag{2.8}$$

where $C_{\Sigma o}$ is the sum of all capacitances connected to node o and q_o is the net charge in node o. The critical voltage of the tunnel junction is expressed as:

$$V_c = \frac{q_e}{2C_{\Sigma o}}. \tag{2.9}$$

We know from the description of the electron trap, that the output voltage reaches its maximum when this voltage reaches the critical voltage of the tunnel junction. Thus, by substituting the critical voltage as in Equation (2.9) into Equation (2.8), the input voltage for which the output voltage reaches its maximum can be expressed as:

$$V_{i,peak} = \frac{q_e}{2C_i} + \frac{kq_e}{C_i} \quad \text{for } k = 0, 1, 2, ... \tag{2.10}$$

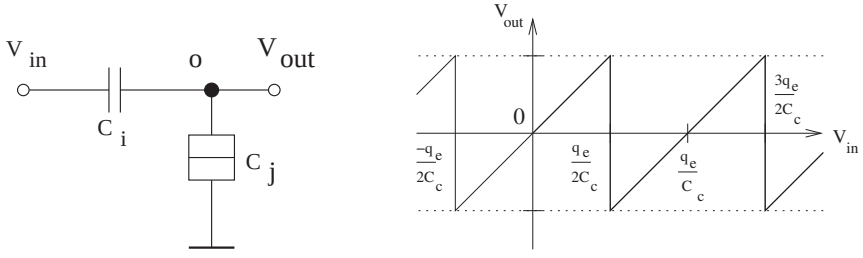
This equation suggests that the period of the electron trap transfer function is dependent only on the magnitude of capacitance C_i, while the capacitance of the tunnel junction has no influence.

To obtain the transfer function in Figure 2.6(b), a SET inverter [25] can be added. The inverter functions as a literal gate. As long as the input is below some threshold value, the output is zero. If the input exceeds the threshold, the output value becomes $'high'$. The full implementation of the PSF block is depicted in Figure 2.8.

The PSF block can be used to perform an analog (representation in amount of electrons) to digital (binary representation) conversion. For every bit in the binary representation a PSF block is needed. In the binary representation the value of every bit is a PSF of the represented value, as one can observe in Figure 2.9. Therefore, by adjusting the period T and the values of a and b every bit can be determined by a PSF block.

2.2.3 Electron Counting Based Addition

Using $MVke$ and PSF blocks as basic building blocks an addition circuit, operating under the EC paradigm, can be designed. The general circuit organization is depicted



(a) Circuit. (b) Transfer function.

Figure 2.7: SET electron trap.

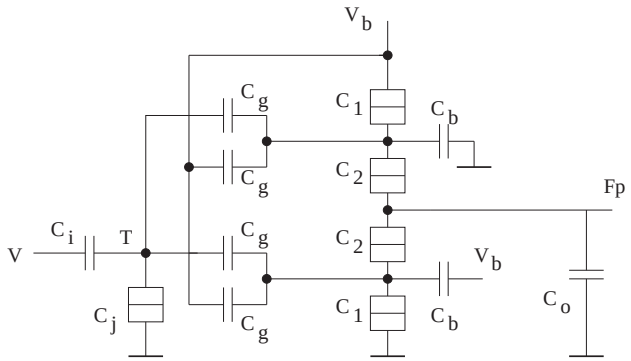


Figure 2.8: PSF building block implementation.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
bit 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
bit 2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
bit 3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 2.9: AD conversion using a periodic symmetric function.

in Figure 2.10. The circuit has two n-bit binary operands, $A = \{a_0, a_1, ..., a_{n-1}\}$ and $B = \{b_0, b_1, ..., b_{n-1}\}$. Every input bit is connected to the V input of a $MVke$ block, of which the built in constant k is set to the weight of the corresponding bit. Thus, if bit a_i or b_i is '1' then 2^i electrons are added to the charge reservoir. In this way, the operands are converted from binary to analog (DA conversion) and at the same time are added in the charge reservoir.

At the second stage of the adder, $n + 1$ PSF blocks are used to perform the AD conversion as explained in the previous section. Subtraction can also be performed using this scheme. The implementation of the $MVke$ block presented above can only remove electrons from the reservoir. For subtraction one also needs to add electrons to the reservoir. Therefore a different implementation of the $MVke$ block would be needed.

2.2.4 Electron Counting Based Multiplication

To perform multiplication, we use the fact that the $MVke$ block is capable of transporting a variable number of electrons depending on the magnitude of the value V. Assume we have again the operands $A = \{a_0, a_1, ..., a_{n-1}\}$ and $B = \{b_0, b_1, ..., b_{n-1}\}$. First operand B is converted to analog and stored in a charge reservoir. This conversion is done in the same way as in the adder scheme. The presence of electrons in the charge reservoir results in some voltage. This voltage is used for the V inputs of the next set of $MVke$ blocks. The bits of A are connected to the E input of the $MVke$ blocks. These $MVke$ blocks are adjusted to move 2^iB electrons, where i is the bit position the $MVke$ block is connected to, and B is simply the value of operand B as present in the intermediate charge reservoir. The general circuit organization is depicted in Figure 2.11.

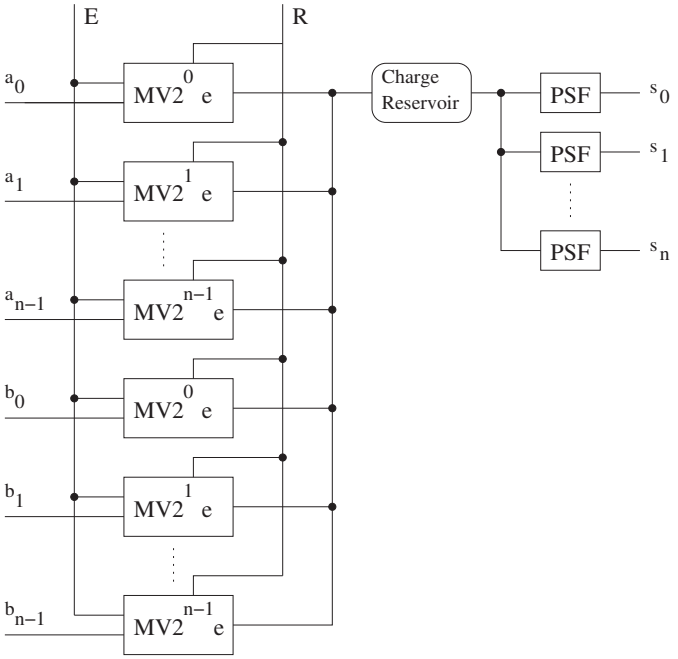


Figure 2.10: Organization of n-bit addition/subtraction circuit.

In this chapter a brief overview of the SET principles were given. The electron counting paradigm was introduced and the building blocks designed so far were presented. As stated before, for the research of this thesis new building blocks had to be designed. For that purpose we introduce in the next chapter a design methodology that allows for the systematic design of SET based building blocks.

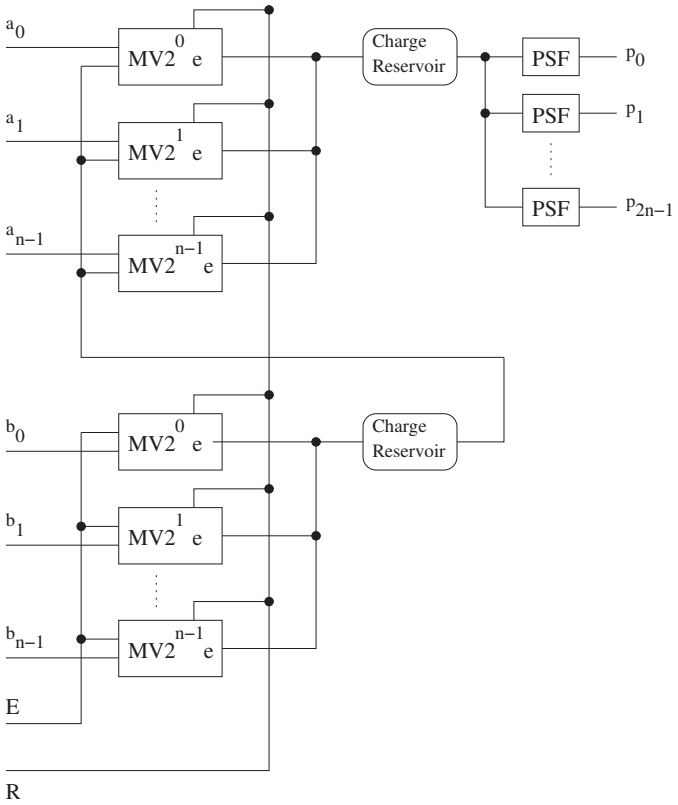


Figure 2.11: Organization of n-bit multiplication circuit.

3 Design Methodology

Designing circuits with SET components has not been done very often. Computer design tools are not available and only a small number of simulators exist. Therefore the design of building blocks is generally not automated. This chapter provides a methodology for finding the circuit parameters of SET based building blocks. The design methodology presented in this section focusses on implementations where values are represented by single or few electrons, like SEEL [24] and the EC paradigm, which was explained in Section 2.2.

This chapter is organized as follows. In Section 3.1 a brief overview of steps of the methodology is given. In Section 3.2 till 3.7 these steps are explained in more detail.

3.1 Methodology Overview

The methodology assumes that there is a circuit topology available which is capable to deliver the targeted behaviour of the building block one wishes to design. Specifying the targeted behaviour is the first step in the design process and follows directly from a certain need. The second step, to come up with a SET topology corresponding to the targeted behaviour, is a more creative process which should not be restricted by a methodology. Therefore these first two steps in the design process are not part of the methodology, but are assumed to be done before starting to use the methodology.

The aim of the methodology is to calculate the circuit parameters given a circuit topology, the corresponding targeted behaviour and some boundary conditions. The design methodology consists of six steps:

1. Derive the characteristic equations for all junctions in the circuit.
2. Given the targeted behaviour and the circuit topology, derive the operation modes.
3. For each mode solve the characteristic equations in order to obtain the relations that have to be satisfied in order to get the corresponding behaviour.
4. For a set of boundary conditions derive the circuit parameters.
5. Determine the upper bounds for correct operation.
6. Verify the design.

In the next sections the steps of the methodology are explained in more detail using an example circuit. Since the $MVke$ building block, presented in Section 2.2.1, is an important block for the research described in this thesis, it is used throughout this chapter to explain the methodology. A more detailed picture of the topology of the $MVke$ block is depicted in Figure 3.1.

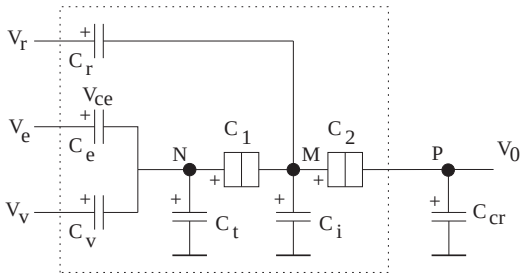


Figure 3.1: $MVke$ block implementation.

3.2 Deriving the Characteristic Equations

The first step of the methodology is to derive the characteristic equations for all junctions of the circuit. The behaviour of a SET circuit is mainly determined by the behaviour of the tunnel junctions. Therefore the behaviour of a SET building block can be described by a set of equations describing the voltages across the junctions in the circuit. For every junction one equation can be derived. In the remainder of this thesis these equations are called the characteristic equations of the building block.

The characteristic equations can be derived by performing the following steps:

- Derive the basic equations describing the charges on the capacitors, tunnel junctions and the circuit nodes.
- For each junction, find a voltage relation to start the derivation. A good voltage relation should at least contain some important input voltage, the voltage across the junction of which the characteristic equation is being derived, and as many other voltages across junctions as possible.
- Specify what parameters are know, what are unknown and what of the unknown parameters should be known (are relevant) to build the circuit. The parameters that are unknown and do not have to be known to build the circuit are non-relevant.
- Starting from the chosen voltage relation, eliminate all the non-relevant parameters.

To derive the basic equations describing the charges on the tunnel junctions they can be treated as capacitors, and therefore the charges can be described as the product of the capacitance of the junction times the voltage across it. For the $MVke$ building block the following basic equations were found.

$$q_1 = C_1 V_1 \tag{3.1}$$

$$q_2 = C_2 V_2 \tag{3.2}$$

$$q_r = C_r[V_r - V_2 - V_0] = C_r[V_r + V_{ce} + V_1 - V_e] \tag{3.3}$$

$$q_i = C_i V_i = C_i[V_2 + V_0] = C_i[V_e - V_{ce} - V_1] \tag{3.4}$$

$$q_v = C_v[V_v - V_t] \tag{3.5}$$

$$q_{ce} = C_e V_{ce} = C_e[V_e - V_t] \tag{3.6}$$

$$q_t = C_t V_t \tag{3.7}$$

$$q_{cr} = C_{cr} V_0 \tag{3.8}$$

The charges on the nodes are described by:

$$q_m = q_2 - q_1 + q_i - q_r = -mq_e + q_{0m} \tag{3.9}$$

$$q_n = q_1 + q_t - q_v - q_{ce} = -nq_e + q_{0n} \tag{3.10}$$

$$q_p = q_{cr} - q_2 = -pq_e + q_{0p}, \tag{3.11}$$

where m, n, p are the number of electrons and q_{0m}, q_{0n}, q_{0p} are the background charges present at nodes M, N, P, respectively. The background charge is a random offset charge induced by stray capacitances and impurities located near the island. They are stated here for correctness, but are omitted in the remainder of this thesis, since as stated before the focus of this thesis is on functional behaviour.

To keep the equations readable the next short notations are introduced,

$$C_{\Sigma n} = C_v + C_e + C_t \tag{3.12}$$

$$C_{\Sigma m} = C_1 + C_2 + C_r + C_i. \tag{3.13}$$

The $MVke$ block contains two tunnel junctions, so two characteristic equations should be derived and for both we need a voltage relation to start with. For the topology of the $MVke$ block there is a voltage relation which meets the conditions of a good starting point for both derivations:

$$V_e = V_{ce} + V_1 + V_2 + V_0 \tag{3.14}$$

This voltage relation contains the voltages across all junctions of the circuit and the input voltage V_e. One could have chosen to include input V_v instead of input V_e for this would not affect the derivation very much.

The next step is to specify what parameters are known, what are unknown, what should be known and what are non-relevant. In general all input voltages are known or should be known, all capacitances and charges on nodes should be known. At last the voltage across the junction of which the characteristic equation is derived, should be known. All other parameters are non-relevant including the output voltages. In Table 3.1 the parameters of the $MVke$ block are specified for the derivation of the junction 1 characteristic equation.

The last step of the derivation of the characteristic equations is to eliminate all non-relevant parameters from the voltage relation. First this is done for junction 1, after which the same is done for junction 2.

known	*unknown*	
	relevant	*non-relevant*
V_v, V_e, V_r, C_{cr}	C_1, C_2, C_r, C_v, C_e, C_t, C_i, q_m, q_n, q_p, V_1	q_1, q_2, q_r, q_v, q_{ce}, q_t, q_i, q_{cr}, V_{ce}, V_t, V_i, V_0, V_2

Table 3.1: Parameters of $MVke$ building block categorized for junction 1.

Junction 1

The elimination of the non-relevant parameters from the start point voltage relation is done by subsequent substitution with the basic equations. It is a matter of diligence and a little mathematical insight and can be done in different ways, though we used the following procedure. First we substituted the non-relevant voltages from the start point voltage relation for capacitances and charges, using Equations (3.1) - (3.8). Second, to eliminate the resulting non-relevant charges we used Equations (3.9) - (3.11). Thus, substitution of the non-relevant voltages from Equation (3.14) resulted in:

$$V_e = \frac{q_{ce}}{C_e} + V_1 + \frac{q_2}{C_2} + \frac{q_{cr}}{C_{cr}} \tag{3.15}$$

Using Equation (3.8), factor q_{cr} was eliminated from Equation (3.15).

$$V_e = \frac{q_{ce}}{C_e} + V_1 + \frac{C_2 + C_{cr}}{C_2 C_{cr}} q_2 + \frac{q_p}{C_{cr}} \tag{3.16}$$

The reader should notice that the elimination of q_{cr} introduced another factor q_2 and therefore it was substituted as first. To eliminate q_2 Equation (3.9) was used but first it was rewritten to eliminate q_i and q_r from it.

$$q_2 = q_m + q_1 - q_i + q_r \tag{3.17}$$

$$q_2 = q_m + [C_1 + C_i + C_r]V_1 + C_r V_r - [C_i + C_r]V_e + [C_i + C_r]\frac{q_{ce}}{C_e} \tag{3.18}$$

Substitution of Equation (3.18) in Equation (3.16) resulted in:

$$\begin{aligned} V_e = {} & \frac{q_{ce}}{C_e} + V_1 + \frac{q_p}{C_{cr}} + \\ & \left[\tfrac{C_2 + C_{cr}}{C_2 C_{cr}}\right] \left[q_m + [C_1 + C_i + C_r]V_1 + C_r V_r - [C_i + C_r]V_e + [C_i + C_r]\tfrac{q_{ce}}{C_e}\right]. \end{aligned} \tag{3.19}$$

Note that another factor q_{ce} was introduced and therefore it was eliminated as last. Elimination of q_{ce} was done using Equation (3.10), which was rewritten to

$$q_{ce} = \frac{C_1 V_1 - q_n + C_t C_e - C_v V_v + C_v V_e}{C_{\Sigma n}} C_e. \tag{3.20}$$

To keep things readable two new short notations were introduced.

$$r_\alpha = \frac{C_2 + C_{cr}}{C_2 C_{cr}}[C_i + C_r] + 1 \tag{3.21}$$

$$r_\beta = \frac{C_2 + C_{cr}}{C_2 C_{cr}}[C_i + C_r + C_1] + 1 \tag{3.22}$$

Substitution of q_{ce} in Equation (3.19) and the usage of these short notations resulted in

$$\begin{aligned} V_e &= \left[r_\beta + \frac{r_\alpha C_1}{C_{\Sigma n}}\right] V_1 + \left[\frac{C_2 + C_{cr}}{C_2 C_{cr}}\right] C_r V_r + \left[\frac{C_t + C_v}{C_{\Sigma n}} r_\alpha - C_{\Sigma a} + 1\right] V_e \\ &\quad - \frac{r_\alpha}{C_{\Sigma n}} C_v V_v + \frac{q_p}{C_{cr}} + \left[\frac{C_2 + C_{cr}}{C_2 C_{cr}}\right] q_m - \frac{r_\alpha}{C_{\Sigma n}} q_n. \end{aligned} \tag{3.23}$$

Taking V_1 to the left side resulted in the first characteristic equation.

$$V_1 = \frac{r_\alpha \frac{C_e}{C_{\Sigma n}} V_e - \frac{C_2 + C_{cr}}{C_2 C_{cr}} C_r V_r + \frac{r_\alpha}{C_{\Sigma n}} C_v V_v - \frac{q_p}{C_{cr}} - \frac{C_2 + C_{cr}}{C_2 C_{cr}} q_m + \frac{r_\alpha}{C_{\Sigma n}} q_n}{r_\beta + \frac{r_\alpha C_1}{C_{\Sigma n}}} \tag{3.24}$$

This equation is accurate but not very easy to use and to understand. Often some simplifications can be applied to the characteristic equations. With some experience and understanding of SET circuits one can most likely find some condition allowing a simplification. For the current case these conditions are $C_2 \ll C_{cr}$ and $C_1 \ll C_{\Sigma n}$. From previous research [22, 9] we know that these conditions hold true, and under these assumptions the characteristic equation of junction 1 becomes:

$$V_1 = \frac{\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_e V_e + C_v V_v + q_n] - C_r V_r - \frac{C_2}{C_{cr}} q_p - q_m}{C_{\Sigma m}}. \tag{3.25}$$

Junction 2

The categorization of the parameters for junction 2 is slightly different than the one presented in Table 3.1, namely V_1 is non-relevant while V_2 is relevant. The derivation of the characteristic equation for junction 2 was done similar as for junction 1. We started with Equation (3.14) and substituted the non-relevant voltages using the basic Equations (3.1) - (3.8).

$$V_e = V_{ce} + V_1 + V_2 + V_0 = \frac{q_{ce}}{C_e} + \frac{q_1}{C_1} + V_2 + \frac{q_{cr}}{C_{cr}} \tag{3.26}$$

Equation (3.20) was used to eliminate the factor q_{ce}.

$$V_e = \frac{-q_n + C_t V_e - C_v V_v + C_v V_e}{C_{\Sigma n}} + \frac{C_{\Sigma n} + C_1}{C_{\Sigma n} C_1} q_1 + V_2 + \frac{q_{cr}}{C_{cr}} \tag{3.27}$$

In order to eliminate q_1 Equation (3.9) was rewritten to

$$q_1 = [C_2 + C_i + C_r] V_2 - q_m + [C_i + C_r] \frac{q_{cr}}{C_{cr}} - C_r V_r. \tag{3.28}$$

Using this equation, Equation (3.27) became:

$$\begin{aligned} V_e &= \frac{-q_n + C_t V_e - C_v V_v + C_v V_e}{C_{\Sigma n}} + \frac{C_{\Sigma n} + C_1}{C_{\Sigma n} C_1}[-q_m - C_r V_r] \\ &\quad + \frac{[C_{\Sigma n} + C_1][C_{\Sigma m} - C_1] + C_{\Sigma n} C_1}{C_{\Sigma n} C_1} V_2 \\ &\quad + \frac{[C_{\Sigma n} + C_1][C_i + C_r] + C_{\Sigma n} C_1}{C_{\Sigma n} C_1} \frac{q_{cr}}{C_{cr}}. \end{aligned} \tag{3.29}$$

The reader should notice that an other factor q_{cr} was introduced and therefore this factor was eliminated as last. Using Equations (3.8) and (3.9) resulted in the following expression:

$$\begin{aligned} V_e &= \frac{-q_n + C_t V_e - C_v V_v + C_v V_e}{C_{\Sigma n}} + \frac{C_{\Sigma n} + C_1}{C_{\Sigma n} C_1}[-q_m - C_r V_r] \\ &+ \frac{[C_{\Sigma n} + C_1]\left\{[C_{\Sigma m} - C_1] + \frac{C_2}{C_{cr}}[C_i + C_r]\right\} + C_{\Sigma n} C_1\left[1 + \frac{C_2}{C_{cr}}\right]}{C_{\Sigma n} C_1} V_2 \\ &+ \frac{[C_{\Sigma n} + C_1][C_i + C_r] + C_{\Sigma n} C_1}{C_{\Sigma n} C_1} \frac{q_p}{C_{cr}} \end{aligned} \tag{3.30}$$

Taking V_2 to the left side resulted in the second characteristic equation.

$$\begin{aligned} V_2 = &\frac{[C_e V_e + C_v V_v + q_n] C_1 + [C_{\Sigma n} + C_1][q_m + C_r V_r]}{[C_{\Sigma n} + C_1]\left\{[C_{\Sigma m} - C_1] + \frac{C_2}{C_{cr}}[C_i + C_r]\right\} + C_{\Sigma n} C_1\left[1 + \frac{C_2}{C_{cr}}\right]} \\ &- \frac{\{[C_{\Sigma n} + C_1][C_i + C_r] + C_{\Sigma n} C_1\}\frac{q_p}{C_{cr}}}{[C_{\Sigma n} + C_1]\left\{[C_{\Sigma m} - C_1] + \frac{C_2}{C_{cr}}[C_i + C_r]\right\} + C_{\Sigma n} C_1\left[1 + \frac{C_2}{C_{cr}}\right]} \end{aligned} \tag{3.31}$$

Again the characteristic equation can be simplified for the conditions $C_2 \ll C_{cr}$ and $C_1 \ll C_{\Sigma n}$, which results in the simplified characteristic equation for junction 2:

$$V_2 = \frac{[C_e V_e + C_v V_v + q_n]\frac{C_1}{C_{\Sigma n}} + q_m + C_r V_r - [C_i + C_r + C_1]\frac{q_p}{C_{cr}}}{C_{\Sigma m}}. \tag{3.32}$$

3.3 Deriving the Operation Modes

For every circuit several modes can be distinguished and the mode of the circuit determines what basic operation is performed in the circuit. For a memoryless circuit the operation mode is only dependent on the input values, while for a circuit containing some memory, both the input values and the inner state determine the mode.

Given a circuit, one could derive the operation modes for every allowed combination of input values and inner states. For simple Boolean circuits this approach might work well, but for larger circuits this might result in too many operation modes. For circuits that contain an analog input this approach is even impossible[1]. But for most circuits one can derive the operation modes intuitively based on the targeted behaviour and the circuit topology.

After the operation modes are derived, one should describe clearly for each mode what the targeted behaviour of the circuit is (i.e. what tunnel events should take place). If necessary one should also mention what the inner state of the circuit should become in terms of voltages and charges.

For our example, the *MVke* building block, we discerned four modes of operation, which are described below.

[1] If the range of an analog input would be divided into subranges, then this approach might be used by deriving modes for every allowed combination of subranges and inner states.

Enabled-mode

The first operation mode of the $MVke$ building block is the enabled-mode. In the enabled-mode, the input V_e is $'high'$ while the other inputs V_v and V_r are $'low'$ and no electrons have tunnelled yet, thus all internal nodes are in a neutral state ($q_n = q_m = q_p = 0$). When the circuit is in the enabled-mode, no tunnel events should take place, but the voltage across junction 1 should be very close to its critical voltage.

Move-mode

The second operation mode is the move-mode, in which both inputs V_e and V_v are $'high'$ and input V_r is $'low'$. The critical voltage of junction 1 should be exceeded so that a certain amount of electrons tunnels from node P to node N. We denote the number of electrons we want to transport, for a certain value of V_v, from node P to node N by k and the actual tunnelled electrons by n. The targeted behaviour can now be described as instability for $n < k$ and stability for $n = k$ ($n > k$ should not be allowed).

Hold-mode

The third mode of operation of the $MVke$ block is the hold-mode, in which is the input V_e is $'high'$ and the other inputs V_v and V_r are $'low'$, like in the enabled-mode. The difference is that now k electrons have tunnelled from node P to node N and these electrons should remain in their position. The targeted behaviour can therefore be described as stability for $n = -k$ and $p = k$.

Reset-mode

The last mode is the reset-mode R, in which only the reset input V_r is $'high'$ and the objective of this mode is that all tunnelled electrons return to their original position, so that the charges on nodes N, M and P all become zero. The targeted behaviour can be described as instability for $n > 0$ and stability for $n = 0$.

3.4 Solving the Characteristic Equation

The third step in the design methodology is to derive for each mode the relations that have to be satisfied in order to get the corresponding behaviour. To do this the critical voltage of each junction is derived, using the method explained in Section 2.1. Substitution of the critical voltage into the characteristic equations results in equations describing the stability of the junctions of the circuit. In these equations subsequently are substituted for each mode the input values, the inner state of the circuit and the specified targeted behaviour which results in the relations we were looking for.

To employ the method of critical voltage, for each tunnel junction the critical voltages should be derived. Using the approach presented in Section 2.1, for the $MVke$ structure

the critical voltages of the two junctions were derived as:

$$V_{c1} = \frac{q_e[C_{\Sigma n} + C_r + C_i + \frac{C_2 C_{cr}}{C_2 + C_{cr}}]}{2\left[[C_1 + C_{\Sigma n}]\left[C_r + C_i + C_1 + \frac{C_2 C_{cr}}{C_2 + C_{cr}}\right] - C_1^2\right]} \tag{3.33}$$

for junction 1 and

$$V_{c2} = \frac{q_e[C_{cr} + C_r + C_i + \frac{C_1 C_{\Sigma n}}{C_1 + C_{\Sigma n}}]}{2\left[[C_2 + C_{cr}]\left[C_r + C_i + C_2 + \frac{C_1 C_{\Sigma n}}{C_1 + C_{\Sigma n}}\right] - C_2^2\right]} \tag{3.34}$$

for junction 2. Assuming $C_1 \ll C_{\Sigma m} \ll C_{\Sigma n}$ these equations describing the critical voltage can be simplified to $V_{c1} = V_{c2} = \frac{q_e}{2C_{\Sigma m}}$. Using this expression in further derivations results in equations which are more comprehensible, but also results in less accuracy.

Using the simplified characteristic equations described in Equations (3.25) and (3.32) the expressions for stability can be written as:

$$\left|\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_e V_e + C_v V_v + q_n] - C_r V_r - \frac{C_2}{C_{cr}} q_p - q_m\right| < V_{c1} C_{\Sigma m} \tag{3.35}$$

$$\left|[C_e V_e + C_v V_v + q_n]\frac{C_1}{C_{\Sigma n}} + q_m + C_r V_r - [C_i + C_r + C_1]\frac{q_p}{C_{cr}}\right| < V_{c2} C_{\Sigma m} \tag{3.36}$$

for junction 1 and junction 2, respectively. Note that we did not substitute any relation for the critical voltages to keep the equations more general. Using the simplified equations for the critical voltage the righthand side of Equations (3.35) and (3.36) would become $V_{c1} C_{\Sigma m} = V_{c2} C_{\Sigma m} = \frac{1}{2} q_e$.

The next step is to substitute, in the stability relations we just derived, for each mode the input values, the inner state of the circuit and the specified targeted behaviour as described in the previous section.

Enabled-mode

In the enabled-mode the voltage across junction 1 should be close to its critical voltage so we took Equation (3.35) and substituted the input and state conditions $V_v = V_r = q_n = q_m = q_p = 0$. Because the voltage across the junction should be close to the critical voltage, we substituted the operator $' <'$ with $' ='$ and subtracted some small value ϵ from the righthand side. The value of ϵ is derived later on. The expression becomes:

$$[C_{\Sigma m} - C_1]\frac{C_e}{C_{\Sigma n}} V_e = V_{c1} C_{\Sigma m} - \epsilon. \tag{3.37}$$

This equation can be used to *derive the value of* C_e. Assuming $C_{\Sigma m}$, C_1, ϵ and V_e are known this equation gives the ratio $\frac{C_e}{C_{\Sigma n}}$.

In this mode the stability relation for junction 2 is not relevant. There are no charges at work and the only active input is V_e which causes a large voltage across junction 1 but only a very small voltage across junction 2. Since we know that the voltage across junction 1 is near the critical voltage, we also know that the voltage across junction 2 is far less than the critical voltage.

Move-mode

For the move-mode the targeted behaviour is instability for $n < k$ and stability for $n = k$. Substitution of these conditions together with input values in Equation (3.35) resulted in

$$\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_e V_e + C_v V_v - nq_e] - \frac{C_2}{C_{cr}} nq_e > V_{c1} C_{\Sigma m} \quad \text{for } n < k \tag{3.38}$$

and

$$\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_e V_e + C_v V_v - nq_e] - \frac{C_2}{C_{cr}} nq_e < V_{c1} C_{\Sigma m} \quad \text{for } n = k. \tag{3.39}$$

Substitution of Equation (3.37) into this expression eliminated both the factors C_e and $V_{c1}C_{\Sigma m}$. This resulted in:

$$\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_v V_v - nq_e] - \frac{C_2}{C_{cr}} nq_e > \epsilon \quad \text{for } n < k \tag{3.40}$$

and

$$\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_v V_v - nq_e] - \frac{C_2}{C_{cr}} nq_e < \epsilon \quad \text{for } n = k. \tag{3.41}$$

From the equations above and from the description of the targeted behaviour, it can be seen that stability occurs if the n transported electrons have compensated for the contribution of the input V_v to the voltage across junction 1. This condition can be stated as follows:

$$\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_v V_v - nq_e] - \frac{C_2}{C_{cr}} nq_e = 0 \tag{3.42}$$

Rewriting this equation and substituting k for n we got:

$$C_v V_v = \left[\frac{C_2}{C_{cr}} \frac{C_{\Sigma n}}{[C_{\Sigma m} - C_1]} + 1\right] kq_e \tag{3.43}$$

from which we can *derive the value for* C_v.

The threshold between stability and instability should be between the last but one and the last tunneling electron. The value ϵ acts as a threshold and should therefore be set to half the contribution of one electron.

$$\epsilon = \frac{1}{2} q_e \left[\frac{C_2}{C_{cr}} + \frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}\right] \tag{3.44}$$

In the enabled-mode, the equation for junction 2 is of not much importance, although due to the tunnel events on junction 1, k electrons tunnelled from node P through junction 2 towards node N. Note that the charge distribution has become $-k$, 0, k for nodes N, M, P, respectively. Substitution of the known values into the stability relation of junction 2, as stated in Equation (3.36), resulted in an upper limit for k but this upper limit is higher than the limit that is derived in the next subsection and is therefore omitted.

Hold-mode

The targeted behaviour for the hold-mode is stability for $n = -k$ and $p = k$. Substitution of these conditions and the input values in Equation (3.35) resulted in the following expression for junction 1.

$$\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_e V_e + -kq_e] - \frac{C_2}{C_{cr}} kq_e < V_{c1} C_{\Sigma m} \tag{3.45}$$

Substitution of Equation (3.37) into this expression, and with a little rewriting this resulted in:

$$k < \frac{2V_{c1}C_{\Sigma m}}{q_e\left[\frac{[C_{\Sigma m}-C_1]}{C_{\Sigma n}} + \frac{C_2}{C_{cr}}\right]} + \frac{1}{2}. \tag{3.46}$$

This equation gives an upper limit for k, the number of electrons we want to remove from the reservoir, but this equation only accounts for junction 1. The derivation for junction 2 resulted in the following expression.

$$kq_e\left[\frac{C_1}{C_{\Sigma n}} + \frac{C_i + C_r + C_1}{C_{cr}}\right] < V_{c2}C_{\Sigma m} + C_e V_e \frac{C_1}{C_{\Sigma n}} \tag{3.47}$$

This equation also gives an *upper limit for k*. Substitution of real values of current implementations showed that this limit is much more tight as it is given by Equation (3.46). In Section 3.6 the limit of Equation (3.47) is elaborated.

Reset-mode

For the reset-mode the targeted behaviour is instability for $n > 0$ and stability for $n = 0$. Substitution of these conditions and the input values resulted in the following two equations for junction 1.

$$-C_r V_r - \frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}} nq_e - \frac{C_2}{C_{cr}} nq_e < -V_{c1}C_{\Sigma m} \quad \text{for } n > 0 \tag{3.48}$$

$$-C_r V_r > -V_{c1}C_{\Sigma m} \quad \text{for } n = 0. \tag{3.49}$$

Again the threshold between stability and instability was placed halfway the last but one and last electron, which resulted for junction 1 in a factor ϵ, equal to the one in Equation (3.44). The resulting equation was:

$$C_r V_r = V_{c1}C_{\Sigma m} - \epsilon. \tag{3.50}$$

In this mode we also have to consider junction 2. One can see that the last electron to tunnel back to the reservoir is the critical one, so we used the conditions $n = 0$, $m = -1$ and $p = 1$ to substitute in the stability equation for junction 2, i.e. Equation (3.36). The targeted behaviour is instability until that last electron has tunnelled from node m to node p.

$$-q_e + C_r V_r - \frac{[C_{\Sigma m} - C_1]}{C_{cr}} q_e < -V_{c2}C_{\Sigma m} \tag{3.51}$$

Placing the threshold halfway between the last but one and the last electron for this equation results in a different factor, which we call ϵ_r.

$$\epsilon_r = \frac{1}{2}\frac{[C_{\Sigma m} - C_1]}{C_{cr}} q_e \tag{3.52}$$

For correct operation the smallest of the factors ϵ and ϵ_r should be chosen. Lets assume for now that ϵ_r is smaller than ϵ. The equation for C_r now becomes:

$$C_r V_r = V_{c2} C_{\Sigma m} - \epsilon_r \tag{3.53}$$

and can be used to *derive the value for* C_r. Reminding the reader that $V_{c2}C_{\Sigma m}$ can be simplified to $\frac{1}{2}q_e$ (see page 22), one should notice that the factor $C_r V_r$, ignoring the small value ϵ, is independent of any circuit parameter. This means that given the voltage of logic signals, including the reset, the value for C_r is fixed.

3.5 Deriving the Circuit Parameters

The fourth step of the methodology is to derive the circuit parameters given a set of boundary conditions. This is done by substituting the values of the boundary conditions into the equations found in the previous step.

Assume we want to implement our example, the $MVke$ block, such that is moves 16 electrons when $V_v = 16mV$ and suppose we have the following boundary conditions:

$$\begin{aligned} V_e('high') &= 16 \text{ mV} \\ V_r('high') &= 16 \text{ mV} \\ C_{cr} &= 10^{-14} \text{ F.} \end{aligned} \tag{3.54}$$

In compliance with previous research, specifically [22, 9], we chose the following parameter values:

$$\begin{aligned} C_1 &= 0.5 \text{ aF} \\ C_2 &= 0.5 \text{ aF} \\ C_{\Sigma m} &= 10 \text{ aF.} \end{aligned} \tag{3.55}$$

In order to satisfy the condition $C_1 \ll C_{\Sigma n}$ we choose $C_{\Sigma n} = 1000$ aF. Substitution of these values in the equations from Section 3.4 gave the results stated in Table 3.2. The second column of the table states what equation was used to calculate the value.

Parameter Value	Equation
$\epsilon = 7.65 * 10^{-22}$ C	(3.44)
$\epsilon_r = 1.52 * 10^{-22}$ C	(3.52)
$C_e = 527$ aF	(3.37)
$C_v = 161$ aF	(3.43)
$C_t = 312$ aF	(3.12)
$C_r = 5.00$ aF	(3.53)

Table 3.2: Calculated parameters for $MVke$ building block.

3.6 Determining the Upper Bounds

In Section 3.4 it was explained how to derive the relations that have to be satisfied in order to get the targeted behaviour. In general this process also delivers some equations describing a limit to the operation. To prevent the circuit from incorrect behaviour, these limits should be evaluated. How this should be done is dependent on the nature of the limit. In general it is a matter of manipulating the equations until some useful and comprehensible equation is found.

For the $MVke$ block two equations (Equation (3.46) and (3.47)) were derived that give an upper limit to the number of electrons that can be moved. As said before, Equation (3.47) gives the lowest upper limit of the two for current applications. Therefore in this section we elaborate on this equation.

Reviewing Equation (3.47) it can be seen that it contains the factors $C_{\Sigma n}$ and C_e. Both factors are dependent on the value of C_v, which in turn is dependent on k through Equation (3.43). Since it is the limit of k we want to determine, these dependencies have to be substituted in Equation (3.47). First of all, Equation (3.43) can be simplified to:

$$C_v = \frac{kq_e}{V_v}. \tag{3.56}$$

This simplification makes it possible to derive the upper limit to k analytical. Without this simplification one would have to use numerical analysis. The simplification does introduce some deviation in the final result, but this proved to be a very small error.

The second simplification is to omit capacitance C_t, which was used to satisfy the condition $C_1 \ll C_{\Sigma n}$. Since we are deriving an upper limit for k, the value of C_v is large and and therefore also C_e and $C_{\Sigma n}$. Thus the condition $C_1 \ll C_{\Sigma n}$ is met and C_t can be assumed to be zero without causing any problems.

Substituting the relation $C_{\Sigma n} = C_e + C_v$ and Equation (3.56) into Equation (3.37) resulted in the following relation between C_e and k.

$$C_e = \frac{V_{c1}C_{\Sigma m} - \epsilon}{[C_{\Sigma m} - C_1]V_e - V_{c1}C_{\Sigma m} + \epsilon}\frac{kq_e}{V_v} \tag{3.57}$$

Using this equation the relation between $C_{\Sigma n}$ and k was described as:

$$C_{\Sigma n} = \frac{[C_{\Sigma m} - C_1]V_e}{[C_{\Sigma m} - C_1]V_e - V_{c1}C_{\Sigma m} + \epsilon}\frac{kq_e}{V_v} \tag{3.58}$$

Substitution of these equations into Equation (3.47) resulted in the final upper limit relation.

$$\frac{C_1 + C_i + C_r}{C_{cr}} kq_e < V_{c2}C_{\Sigma m} + [V_{c1}C_{\Sigma m} - \epsilon]\left[\frac{C_1}{C_{\Sigma m} - C_1}\right] - \frac{[C_{\Sigma m} - C_1]V_e - V_{c1}C_{\Sigma m} + \epsilon}{[C_{\Sigma m} - C_1]V_e} C_1 V_v \tag{3.59}$$

The reader is reminded that this equation is not exact due to simplifications. One should always check an implementation using Equation (3.47) or by simulation.

Note that this limit is derived for the $MVke$ block as a stand alone circuit element. When connecting this block to other circuits things changes. For example, the charge reservoir connected to the output of the $MVke$ block might be connected to other blocks that also remove electrons from the reservoir, resulting in a larger positive voltage on the output as accounted for in the equations above. This causes the limit to decrease.

For the $MVke$ block, using the parameters from Section 3.5, we obtained an upper limit for k of 529.

3.7 Verifying the Design

The last step of the methodology is the verification of the design by means of simulation. Throughout this thesis the simulator SIMON [36, 1] is used but another available SET simulator is MOSES [2].

The example circuit used throughout this chapter, the $MVke$ block, was simulated using the parameters calculated in Section 3.5. The simulation indicated that the circuit functions correctly and moves exactly 16 electrons from the reservoir for $V_v = 16mV$. Through simulation we also determined the upper limit for k as 531. The difference with the result obtained in the previous section, by using Equation (3.59), is due to the simplifications used in deriving that equation.

In this chapter we introduced a design methodology that allows to derive circuit parameters in an analytical way, assuming the topology and targeted behaviour are known. In the following chapter we utilize this method for the design of the missing EC element, the conditional move block.

4 The Conditional Move Block

The Electron Counting (EC) paradigm is a novel concept of computation which was first introduced in [9] and described more extensive in [22]. The basic idea of the EC paradigm is to represent integer numbers as an amount of electrons in a charge reservoir. Using the EC paradigm, circuits have been designed to compute addition and multiplication [10], which were reviewed in Section 2.2.

In theory, the EC addition and multiplication circuits perform very well in regard to speed and area. But from a practical point of view there are some limitations to the applicability of these circuits. As previously stated in Chapter 1, the operand size of these EC based adders and multipliers is limited due to two aspects. First, the EC building blocks can transport only a limited amount of charge correctly. Second, the delay of the circuit is proportional to the operand size. To overcome these limitations, computation can be done in high radix.

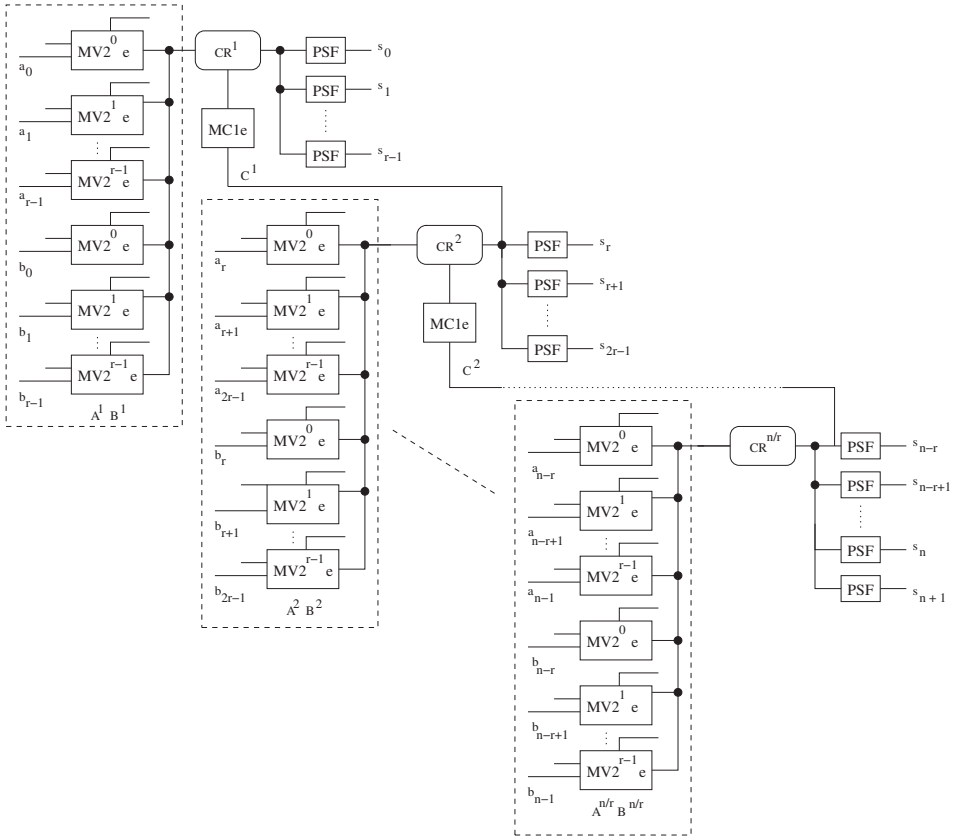


Figure 4.1: Organization of high radix n-bit addition circuit.

A high radix addition organization was proposed in [22], which is depicted in Figure 4.1. The n-bit addition was divided into r-bit slices resulting in $\lceil \frac{n}{r} \rceil$ slices. These slices are similar to the EC adder as reviewed in Section 2.2. The only difference is that an output carry in the form of one electron transport is generated in all but the last slice, and is subsequently added to the charge reservoir of the next slice.

To generate this output carry a new type of building block is needed. This building block is called the 'Move Conditional 1 electron' block or $MC1e$. The block senses the charge in the reservoir and when the number of electrons in the reservoir exceeds a threshold, one electron is removed from the charge reservoir of the next slice. There is no implementation as-of-yet for the $MC1e$ block in previous work. In this chapter we propose implementations for a more generalized version, i.e., the $MCke$ block that can conditionally move k electrons instead of one.

The remainder of this chapter is organized as follows. Section 4.1 presents five potential implementations of the $MCke$ block. Section 4.2 compares the different implementations and argues which design is most suitable for high radix EC based addition and multiplication. Section 4.3 presents a 6-bit radix-8 adder based on the selected $MCke$ block.

4.1 $MCke$ Implementations

The $MCke$ block, as depicted in Figure 4.2, has two Boolean inputs (E, R) which are controlling the behaviour and one analog input (V). If the input E (enable) is high and R (reset) is low and the number of electrons in reservoir 1 exceeds a certain threshold then the block removes k electrons from reservoir 2. If R is high while E is low, the electrons which were removed from reservoir 2 are returned.

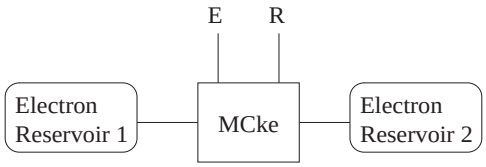


Figure 4.2: $MCke$ building block.

The charge reservoir on the input (electron reservoir 1) is connected via a buffer or an OpAmp. The reason for this is that without such a buffer or OpAmp there would occur dynamic feedback effects which influence the voltage on the charge reservoir. Note that the charge reservoir on the input forms a part of an EC adder. The PSF blocks switch on the voltage across the charge reservoir and would malfunction due to the dynamic feedback. We remark here that OpAmps can potentially be build using a hybrid FET-SET technology [29, 7, 18].

The OpAmp can be utilized to amplify the voltage on the first charge reservoir. Doing so would be beneficial to the design of the next part connected to the OpAmp. To detect smaller voltages, higher accuracy in the component values is needed. Additionally, dealing with small voltages makes the relative influence of dynamic feedback more troublesome.

On the other hand, using the OpAmp as a simple buffer gives us some advantages in simulation. Although the library of SIMON contains an OpAmp, simulating a circuit containing an OpAmp, which on the output is connected to SET circuitry, often failed to give proper results. While simulating such a circuit some random effects seemed to occur. When using the OpAmp as a buffer (amplification factor is one) most of the time, the simulation gave proper results and the circuit could be simulated at once. To simulate a circuit containing amplification factors larger than one, we used a strategy of partitioning. A cut was made in the circuit after each amplifier. The circuits connected to original inputs were simulated first while the output of the amplifiers was measured and stored. The measured outputs of the amplifiers were then feeded back into the next parts of the circuit, at the points which were originally connected to the output of the amplifiers.

Since there has not been designed, to our best knowledge, a hybrid FET-SET OpAmp yet, we did not know what amplification factor would be realistic or possible. Therefore we chose what suited best considering the current conditions.

One can think of an $MCke$ building block as consisting of two parts. The first part detects whether the number of electrons in the charge reservoir connected to the $MCke$ block's input exceeds the threshold value, while the second part is responsible for removing k electrons from the reservoir connected to the output. We remind that the number of electrons in a charge reservoir contributes linearly to the voltage present on the corresponding circuit node. Therefore a threshold value to the number of electrons corresponds with a threshold to the voltage on the circuit node of the charge reservoir. For both the first and the second part, two possible implementations are presented. For the special case $k = 1$ the two parts can be merged into a single part. In other words, a total of five possible implementations of the $MCke$ block are presented.

4.1.1 Implementation I1

The first $MCke$ implementation (I1) is depicted in Figure 4.3. It utilizes two $MVke$ blocks. The first $MVke$ block was set to $k = 1$ and the capacitors C_e, C_v and C_t were chosen such that when V_v exceeds the threshold voltage one electron tunnels. This tunnel event results in a high voltage on capacitor C_{cr} which is the input for the second $MVke$ block. This second $MVke$ block was adjusted to move k electrons when the input equals the voltage resulting from one electron on capacitor C_{cr}.

Note that the buffer in between the two $MVke$ blocks is necessary in order for the second block to move more than one electron. If there would not be any buffer, the charge of one extra electron on node N' would always fully compensate for the charge of the one extra electron on node P. Therefore the transport of electrons would stop immediately after the first electron tunnelled from node P' to node N'.

For each of the five implementations an instance was simulated. The following parameters were chosen in compliance with previous work [9, 22] and these apply to all five implementations. All charge reservoirs, except those inside the $MCke$ block, have a capacitance of $10^{-14}F$. All logic signals (enable, reset and the inputs of the adder) have a $'high'$ value of $16mV$ and the threshold voltage V_{vth} was set to $0.12mV$. All implementations were simulated for different values of k, except implementation I3 which can

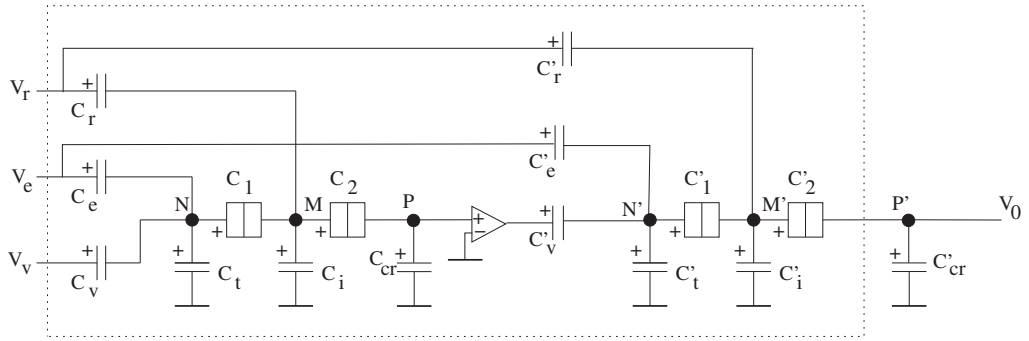


Figure 4.3: $MCke$ building block implementation I1.

move only one electron. All parameter values and simulation results presented in this section correspond to $k = 1$.

For implementation I1 the following parameter values were chosen. The value for the charge reservoir at node P was set to $10^{-16}F$ which is a hundred times smaller than all other charge reservoirs. The reason for this was to create a larger voltage on that node. A drawback of this approach is that the condition $C_{cr} >> C_2$ is not valid anymore. Therefore the simplified equations describing the $MVke$ block could not be used anymore (see Section 3.2). Other values that were chosen are $C_{\Sigma m} = 10aF$ and $C_1 = C_2 = 0.5aF$, which are all conform previous implementations of the $MVke$ block.

For deriving the values of C_v and C_e the equations from Section 3.4 could not be used. In contrast to previous use of the $MVke$ block, the enable input is not alone setting the voltage across junction 1 close to its critical voltage. The critical voltage should be reached when V_e is $'high'$ and when V_v is the threshold voltage. Substitution of these conditions in Equation (3.24) results in the following expression.

$$\frac{\frac{r_\alpha}{C_{\Sigma n}}[C_e V_e + C_v V_v]}{r_\beta + \frac{r_\alpha C_1}{C_{\Sigma n}}} = V_{c1} \tag{4.1}$$

The value for C_v should be chosen as follow. We know that the number of electrons which are transported by the $MVke$ block is related to the input voltage V_v. This relation is described by Equation (3.43). In this case, though, we do not want more than one electron to tunnel. Therefore there is a maximum value for C_v, such that at the maximum input voltage V_{vmax} still one electron tunnels. The (simplified) equation describing this is:

$$C_{vmax} = \left[\frac{C_2}{C_{cr}}\frac{C_{\Sigma n}}{[C_{\Sigma m} - C_1]} + 1\right]\frac{(k+1)q_e}{V_{vmax} - V_{vth}}. \tag{4.2}$$

This maximum value for C_v might be relatively large. A smaller value might be chosen, but this increases the accuracy in component value needed for correct operation. Assuming the maximum input voltage is $0.22mV$, the maximum value for C_v would be $4700aF$. Since this is rather large, C_v was chosen to be $400aF$, and with $C_{\Sigma n} = 1000aF$ this resulted in $C_e = 529aF$. Using the normal equations for the $MVke$ block we calculated $C_t = 709aF$, $C_r = 5aF$ and $C_i = 4aF$.

The second $MVke$ block was implemented as described in Chapter 3. The following values were calculated: $C_v = 100aF$, $C_e = 527aF$, $C_t = 372aF$, $C_r = 5aF$, $C_i = 4aF$ and $C_1 = C_2 = 0.5aF$.

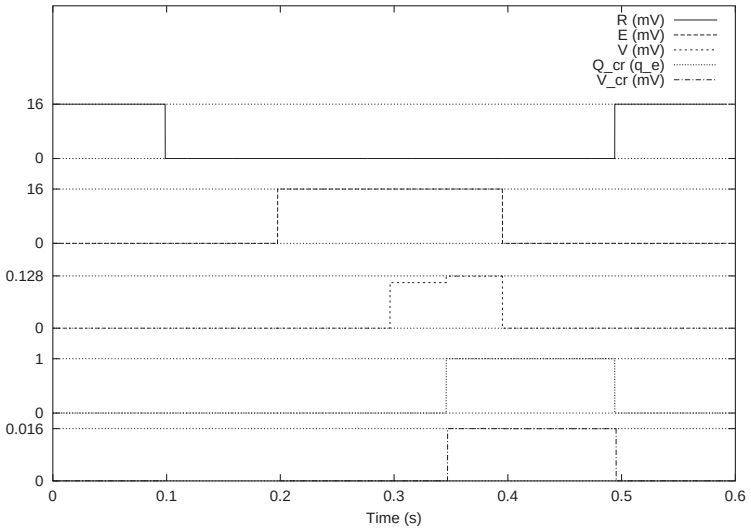


Figure 4.4: Simulation results for $MCke$ implementation I1.

The simulation results for implementation I1 are presented in Figure 4.4. The top three bars represent the inputs R, E and V while the bottom two bars represent the charge present in the output reservoir and the voltage across C_{cr}, implementing the reservoir. The input V starts zero and switches during the enabled period to $0.112mV$, which corresponds to the voltage induced by seven electrons in a charge reservoir with a capacitance of $10^{-14}F$. After a while input V switches to $0.128mV$, corresponding to eight electrons. The threshold is exceeded during this increase and one can see that at that moment the $MCke$ block removes one electron from the output reservoir. Since SIMON is not able to provide accurate delay figures, the time in the plot is fictitious and does not give any indication of the delay of the circuit. This applies to all simulation results presented in this thesis.

4.1.2 Implementation I2

The $MCke$ block has to determine whether some condition was met. Since we work with charges on capacitors this means it has to compare the input voltage with some threshold voltage. A threshold gate is a device which performs such an operation and could therefore be utilized to implement the first stage of a $MCke$ block. SET technology offers great possibilities to implement threshold gates [31, 19]. The threshold gate implementation proposed in [23] was used to design implementation I2 and the resulting circuit is depicted in Figure 4.5. The only difference between implementation I2 and I1 is that the first stage was replaced with a threshold gate.

Implementation I2 was simulated using the parameters as described above. Thus the threshold voltage was again $0.12mV$. The enable signal, which was set to $16mV$

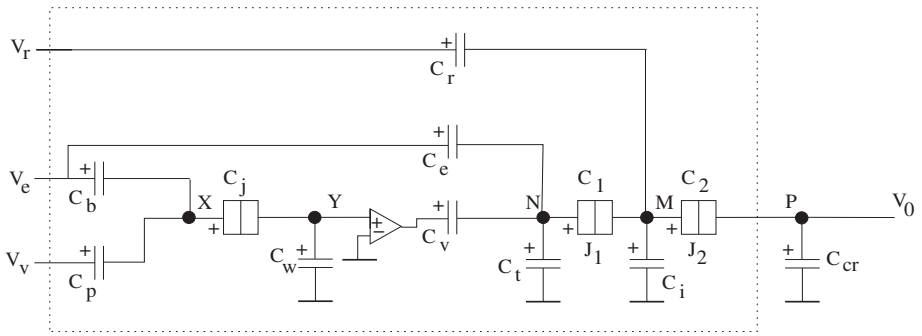


Figure 4.5: $MCke$ building block implementation I2.

was used as the biasing voltage of the threshold gate. The output capacitor C_w of the threshold gate was chosen to be $10aF$ while the junction capacitance C_j which was chosen $0.5aF$. Choosing C_w larger would cause the output voltage to decrease, while choosing C_w smaller would cause the critical voltage of the junction to increase, which in turn would cause the required accuracy for the input to increase. The optimal value for the input capacitance was calculated to be $10aF$. The equations describing the threshold gate are stated in Appendix A. Using these we calculated the biasing capacitance C_b as $19.9aF$. The voltage swing on the output of the threshold gate was calculated as $15mV$. The bias voltage on the output of the threshold gate, caused by capacitance division was $0.5mV$, which is small enough.

The second stage was designed in the same way as it was done for implementation I1. Since the threshold gate has a higher output voltage some parameter values were changed. The following values were calculated: $C_v = 10.4aF$, $C_e = 527aF$, $C_t = 463aF$, $C_r = 5aF$, $C_i = 4aF$ and $C_1 = C_2 = 0.5aF$.

Simulation indicates that implementation I2 of the $MCke$ block functions correctly too. The simulation results are therefore the same as the one depicted in Figure 4.4.

4.1.3 Implementation I3

For the special case when the $MCke$ block only has to move one electron, it is possible to implement this block by using only one $MVke$ building block. Implementation I3, which actually is not an $MCke$ block but an $MC1e$ block, is depicted in Figure 4.6.

The design of this block was done in the same way as the first stage of implementation I1. Because of the absence of the second stage we had to deal with some feedback from the charge reservoir. The voltage on the charge reservoir varies between some bounds and causes dynamic feedback. This voltage on the charge reservoir has effect on the voltage across junction 1 and therefore influences the threshold voltage. To overcome this problem, we had to make sure that the contribution to the junction voltage of the input voltage is much bigger than the contribution of the feedback. In other words, we have to make sure that the input voltage is large enough. Therefore the OpAmp, connecting the first charge reservoir to the $MC1e$ block, was set to amplify a hundred times. Using the methodology and the equations from Section 4.1.1 we calculated the following values: $C_v = 10aF$, $C_e = 525aF$, $C_t = 465aF$, $C_r = 5aF$, $C_i = 4aF$ and

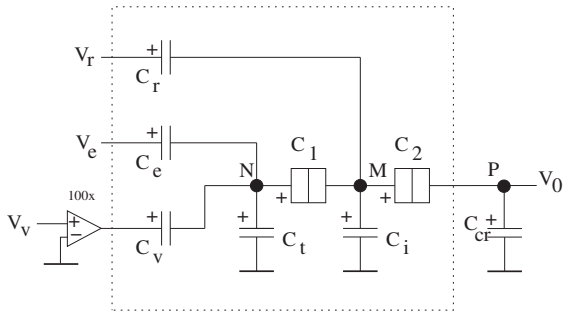


Figure 4.6: $MC1e$ building block implementation I3.

$C_1 = C_2 = 0.5aF$.

Again a simulation was performed, which indicated that this implementation functions correctly. The simulation results are the same as the one depicted in Figure 4.4.

4.1.4 Implementation I4

The first two implementations both needed a buffer between the two stages. Since buffers cause delay and cost area they should be avoided as much as possible. The need for the buffer is due to the way the $MVke$ block of the second stage was implemented. To overcome this problem a modified version of the $MVke$ block was devised. The modified implementation, called the $MMVke$ block, is depicted in Figure 4.7.

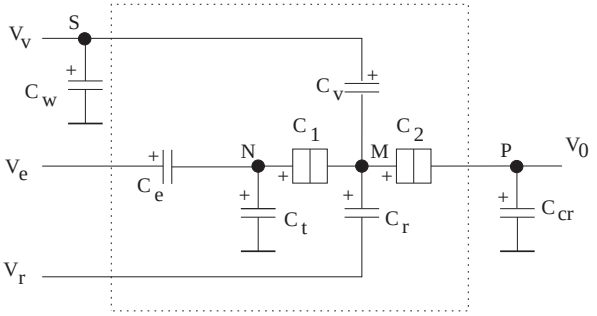


Figure 4.7: $MMVke$ building block implementation.

The $MMVke$ building block functions slightly different than the $MVke$ block does. The enable signal V_e is used to set the voltage across junction 2 close to its critical voltage. The voltage across junction 1 stays below its critical voltage. The driving input V_v is connected to the central node M. If V_v increases from zero to some positive value, the critical voltage of junction 2 is exceeded and an electron tunnels from node P to node M. Consequently, this causes the voltage across junction 1 to exceed its critical voltage and an electron tunnels from node M to node N. This process of two tunnel events repeats itself until the voltage across junction 2 has dropped below the critical voltage again. The number of electrons that tunnels is proportional to the voltage V_v.

Actually the input is not voltage driven but charge driven. It is assumed that the

input is connected to some charge reservoir C_w. The presence of charge on the reservoir results in a voltage V_v, so the statements above remain valid. In describing the functionality mathematical, though, we did not use this voltage but only the charge on the reservoir. A full mathematical description of the $MMVke$ block is given in Appendix A. We suffice with stating the relation between the input and output. Assuming the voltage V_e is $'high'$ while $V_r = 0$ and the number of electrons present on the input charge reservoir equals s, the following relation was found.

$$\frac{C_v}{C_v + C_w} s = \left[\frac{C_1}{C_{\Sigma n}} + \frac{C_{\Sigma m} - C_2}{C_{cr}} \right] k \tag{4.3}$$

The number of electrons removed from the output reservoir is k. As one can see, k is proportional to s but also depending on a lot of other parameters. While designing a block using the $MMVke$ block one should carefully consider all parameter values.

The first $MCke$ building block based on the $MMVke$ block is depicted in Figure 4.8. The first stage is a $MVke$ building block. When the input voltage V_v exceeds the threshold this block removes one electron from node S resulting in the removal of k electrons from the output charge reservoir by the second stage, the $MMVke$ block.

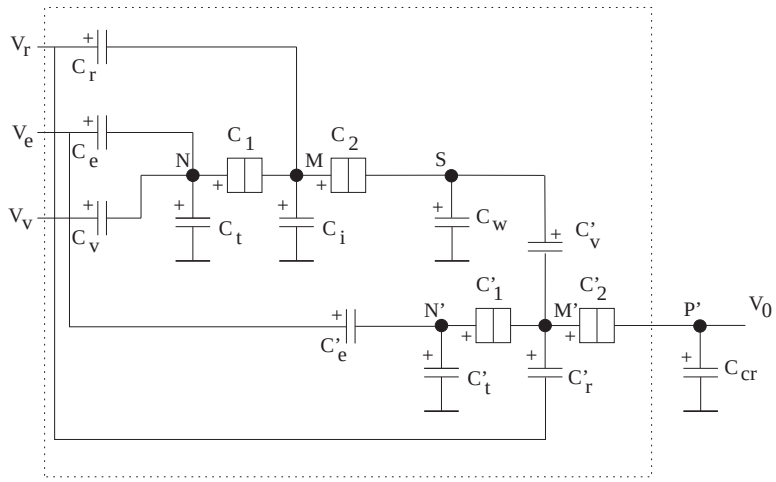


Figure 4.8: $MCke$ building block implementation I4.

This implementation was also simulated for a input threshold voltage of $0.12mV$ and $k = 1$. Corresponding to all previous designs the output charge reservoir has a capacitance of $10^{-14}F$ and all logic signals have a $'high'$ value of $16mV$. We chose $C_w = 50aF$, $C'_v = 2aF$ and C'_2 to be small, that is $0.5aF$. Through calculation we derived $C'_1 = 7.43aF$ which results in $k = 1.5$. Since only discrete electrons can tunnel, and no partial electrons, it is good to adjust k to be about a half too large. This makes the design more robust and allows for some deviation in the component values. It was also calculated that C'_r should be $5aF$, as could be expected.

The values for the first stage were derived in the same way as done in Section 4.1.1. The following values were calculated: $C_v = 400aF$, $C_e = 530aF$, $C_t = 70aF$, $C_r = 5aF$, $C_i = 4aF$ and $C_1 = C_2 = 0.5aF$.

Simulation indicates that implementation I4 of the $MCke$ block functions correctly too. The simulation results are the same as the one depicted in Figure 4.4.

4.1.5 Implementation I5

The last implementation of the $MCke$ block is based on the utilization of threshold gate in the first stage and an $MMVke$ block in the second stage and it is depicted in Figure 4.9. The design process for both stages is the same as described in the previous sections. Using the same assumptions as for the previous implementations the following parameter values were calculated : $C_p = 10aF$, $C_b = 6.76aF$, $C_j = 0.5aF$, $C_w = 50aF$, $C_v = 2aF$, $C_1 = 7.43aF$, $C_2 = 0.5aF$, $C_e = 200aF$, $C_t = 100aF$ and $C'_r = 5aF$.

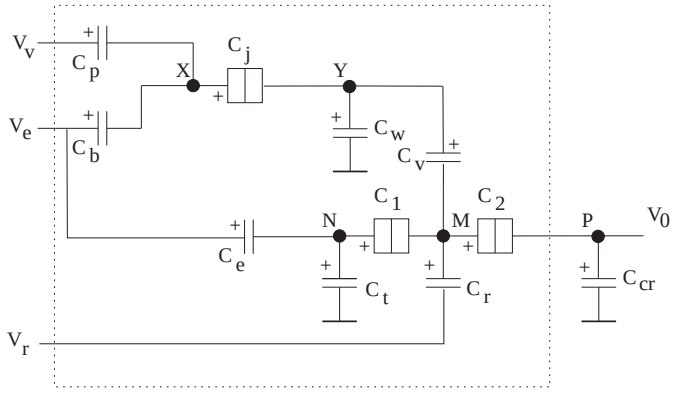


Figure 4.9: $MCke$ building block implementation I5.

This building block was simulated for different values of k too and all simulations indicates that implementation I5 works correctly.

4.2 Comparison of the Implementations

In the previous sections five different implementations of the $MCke$ block were presented. In this section they are compared and the main differences are stated.

In all implementations the first stage was created using either a threshold gate or an $MVke$ block. The threshold gate seems to have a lot of advantages over the $MVke$ block. The first advantage is the larger output swing of the threshold gate. Assuming the same input conditions and the same junction capacitances, the output capacitance of the threshold gate can be about ten times smaller than that of the $MVke$ block. Additionally, a ten times smaller capacitance results in a ten times higher output voltage, which is important since it reduces the relative influence of the dynamic feedback through the output. The larger output voltage of the threshold gate also implies that the next gate will be faster. It also provides more robustness because the design requires a lesser accuracy for the component values.

The second advantage of the threshold gate is the greater input range under which the $MCke$ block functions correctly. For the $MVke$ block, the input is not allowed to exceed

twice the threshold voltage, since this would cause an extra electron to tunnel from the intermediate charge reservoir and k extra electrons from the final charge reservoir. For the threshold gate, the input can be much higher than the threshold voltage. How much higher is depending on the exact implementation. In implementation I2 the input voltage can be up to 500 times higher than the threshold voltage.

The third and last advantage lies in the fact that the threshold gate requires half the number of components the $MVke$ block does. Area, though, is not only determined by the number of components but also by the size of the components. Comparing the size of the total capacitance we find that the threshold gate requires about 27 times less capacitance as the $MVke$ block.

Considering these advantages we conclude that the threshold gate is the best candidate for constructing the first stage of the $MCke$ block.

For the second stage we have the choice between either the $MVke$ block and the modified version of it, the $MMVke$ block. One drawback of the $MVke$ block is the need for a buffer. Since such a buffer has not been designed yet, we cannot give an exact measure to this drawback, but a buffer causes extra delay, cost extra area and consumes extra power.

Concerning the dynamic feedback through the output, the $MMVke$ is a bit more sensitive than the $MVke$ block. The input of the $MMVke$ block is connected to the intermediate node, which is closer to the output than is the input of the $MVke$ block. Therefore the influence of some voltage on the output towards the input, due to capacitive division is smaller at the $MVke$ block. But the $MMVke$ block is more flexible in choosing the parameters, which allows the capacitor values to be optimized for reducing the effect of the dynamic feedback.

The output range of the $MMVke$ building block is more limited. Exact numbers are dependent on the specific situation, but for the input conditions used in this chapter we calculated the following. The $MVke$ block can move up to about 500 electrons, while the $MMVke$ block can move up to about 32 electrons. Also moving electrons using the $MMVke$ block to a reservoir which is already filled with a significant amount of electrons, requires more precision in parameter values as is the case using the $MVke$ block.

Both blocks needs about the same area and consume about the same energy, disregarding the buffer. The $MVke$ block has a smaller delay than the $MMVke$ block. Though, taking the buffer into account, it is expected that the area cost, the energy consumption and the delay when using the $MVke$ block, exceeds those values when using the $MMVke$ block.

Summing up all aspect concerning the last stage, the $MMVke$ block is preferred since it has a lower area cost, energy consumption and delay. The limited output range of the block remains a point of concern. Therefore in some situations the $MVke$ block might be preferred after all.

For the special case $k = 1$ implementation I3 was devised. The simplicity of the implementation is an advantage, but there are also some drawbacks to this approach. In the first place, this implementation is quite sensitive to dynamic feedback through

the output. This could be resolved by increasing the amplification factor of the OpAmp which is in front of the $MCke$ block. But this amplification itself is the second drawback. Amplification is more difficult than just buffering and it is expected that amplification causes a larger delay and cost more area.

Summarizing we can state that implementations I2 and I5 should be preferred over the others also when $k = 1$, since the threshold gate is the best choice to build the first stage of the $MCke$ block. In Table 4.1 the area, delay energy consumption of both implementations are stated, although for implementation $I2$ the required buffer is not included.

Building Block	*Area*	*Delay*	*Energy*
Implementation $I2$	11 elements	1.55 ns	10.3 meV
Implementation $I5$	10 elements	2.13 ns	10.6 meV

Table 4.1: Area, delay and consumed energy of $MCke$ implementations.

When accounting for the delay of the buffer, the delay of Implementation I2 most likely exceeds the delay of implementation I5. OpAmp buffers are expected to consume large amounts of energy compared to SET circuits, thus the total energy consumed by Implementation I2 also exceeds that of Implementation I5. Implementation I5, on the other hand, is more sensitive to dynamic feedback through the output, causing the number of electrons in the output charge reservoir to be bound to a certain limit.

Concluding, Implementation I5 is better at first glance because it is faster, smaller and uses less energy. But when the number of electrons in the output reservoir exceeds the limit it can not be used and Implementation I2 has to be used instead.

4.3 6-bit Radix-8 Adder

In the previous sections we proposed five implementations of the $MCke$ block and selected two out of these. The $MCke$ block was designed because it is needed to implement the EC high radix addition scheme. Therefore in this section we present the complete implementation and simulation results for a 6-bit radix-8 adder. The organization of this circuit is shown in Figure 4.10. All parameters of the EC adder are chosen as in [9, 22]. All charge reservoirs have a capacitance of $10^{-14}F$. All logic signals (enable, reset and the inputs of the adder) have a $'high'$ value of $16mV$.

A carry must be generated when the reservoir contains eight or more electrons. Therefore the threshold voltage of the $MCke$ block should be the voltage induced by the charge of 7.5 electrons on the capacitance of the first charge reservoir. So the threshold voltage V_{vth} is $0.12mV$, which is exactly the threshold voltage we used in Section 4.1. Therefore when building the high radix adder, the parameter values presented in Section 4.1 were used for the $MCke$.

The 6-bit radix-8 adder was simulated using both implementation I2 and I5. The results for both simulations are the same and are depicted in Figure 4.11. The adder

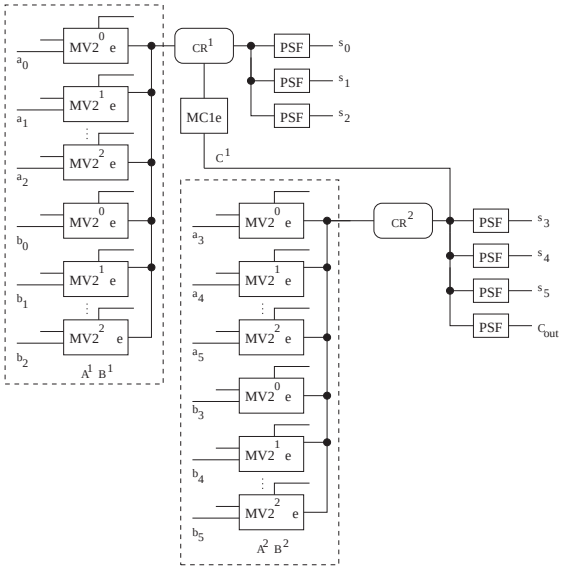


Figure 4.10: Organization of 6-bit radix-8 adder.

was simulated with six different input values. The first three values were such that the second charge reservoir would contain no electrons, while at the last three values it would contain the maximum number of electrons. The reason for this is to check whether the feedback through the output would cause a fault to happen. For both input value sets the following test was done. First the values were chosen such that the first charge reservoir would contain seven electrons, which is just below the threshold. In the second step the charge reservoir would contain 8 electrons, and a carry should be generated. In the last step the first charge reservoir would contain the maximum number of electrons. In this case there should still be one electron removed from the second charge reservoir. As one can see, the high radix adder and the $MCke$ block functions correctly.

In this chapter we discussed the implementation of the $MCke$ building block required for high radix addition EC based addition. Five potential implementations have been proposed and a 6-bit radix-8 adder has been implemented and verified by means of simulation. In the next chapter we address SET based computation of periodic symmetric functions and its utilization in addition related arithmetic operations.

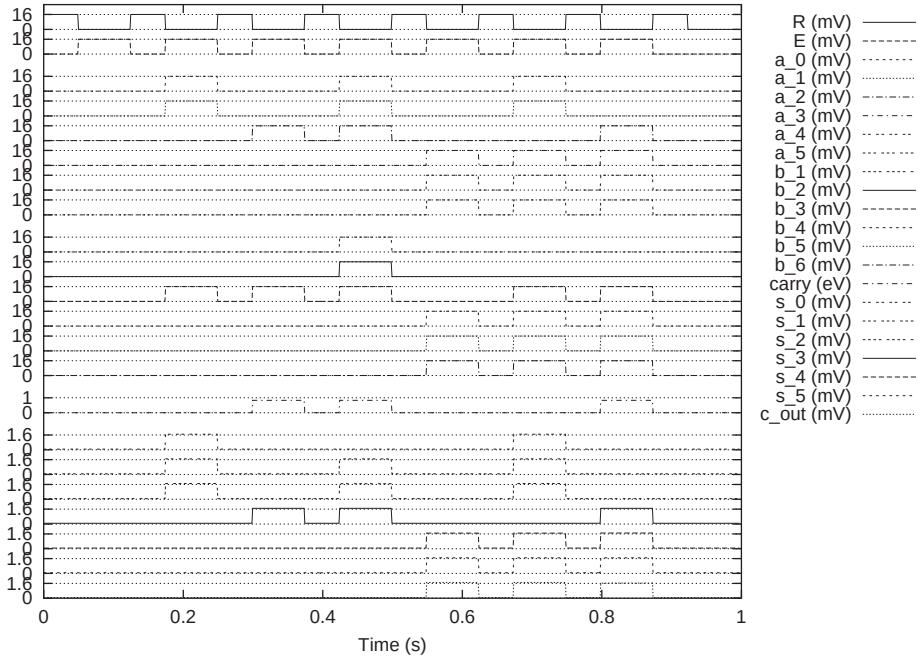


Figure 4.11: Simulation results for 6-bit radix-8 adder.

Periodic Symmetric Function Based Computation

5

Many arithmetic operations can be seen as a Periodic Symmetric Functions (PSFs), more precisely each output bit can be described as a generalized periodic symmetric function of its inputs. In Section 2.2.2 the PSF building block was presented which performs a periodic symmetric function. The PSF block was designed to operate under the Electron Counting paradigm and was used to convert an analog value to the digital domain.

The PSF building block, with a number of adjustments, can be used to perform any arithmetic operation that can be described as a generalized PSF. The resulting circuit is a depth-1 network and it is suitable for a broad range of applications including addition, parity check, Block Save Addition (BSA), etc.

This chapter is organized as follows. In Section 5.1 the subject of symmetric functions is introduced. In Section 5.2 a multiple input PSF building block is proposed. Using this new PSF block, in Section 5.3 as example a 3-bit PSF based addition scheme is designed and simulation results are presented. In Section 5.4 a second example, a block save adder, is implemented using the PSF scheme and this circuit is simulated too. In Section 5.5 a buffered version of the PSF addition scheme is proposed, which allows the adder to be utilized in a network of SEEL gates. This chapter is concluded with some practical considerations in Section 5.6.

5.1 Preliminaries

A function on n variables is symmetric if and only if for any permutation σ of $< 1, 2, ..., n >$, $F_s(x_1, x_2, ..., x_n) = F_s(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$. In other words, a symmetric function is independent on the order of the operands. Addition and multiplication are symmetric functions since $x_1 + x_2 = x_2 + x_1$ and $x_1 * x_2 = x_2 * x_1$.

A Boolean function has operands that are either '0' or '1'. Thus a Boolean symmetric function depends on the number of 'ones' in the input. Moreover, a Boolean symmetric function entirely depends on the sum of its input values: $F_s(x_1, x_2, ..., x_n) = F_s(\sum_{i=1}^{n} x_i)$. A Boolean function is usually described by its truth table. Now, a Boolean symmetric function can be described by a vector $v = v_0 v_1 ... v_n$ where v_i is the output of F_s when the sum of the inputs is i.

A generalized symmetric function $F_g(X)$ is a function that depends on the weighted sum of its inputs $X = (\sum_{i=1}^{n} x_i w_i)$, where w_i is the weight of input x_i. A periodic symmetric function is a symmetric function for which there exists a period T such that $F_s(X) = F_s(X + T)$. A PSF is completely defined by the constants a, b and T, where a is the first positive transition and b is the first negative transition (see Figure 5.1).

Many arithmetic operations can be described as generalized symmetric functions. Assume, for example, the binary addition of operands $\{a_2, a_1, a_0\}$ and $\{b_2, b_1, b_0\}$ result-

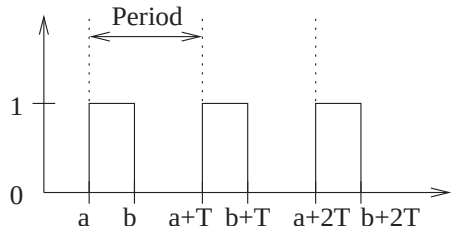


Figure 5.1: Periodic Symmetric Function.

X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
s_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
s_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
s_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Table 5.1: Symmetric functions $F_s(X)$ for sum bits of 3-bit addition.

ing in a sum $\{s_3, s_2, s_1, s_0\}$. A weight $w_k = 2^k$ can be assigned to every a_k, b_k such that $X = \sum_{k=1}^{n} 2^k(a_k + b_k)$. Each sum bit s_i can be described as a symmetric function of X as shown by Table 5.1.

From the table a periodicity can be observed for all four symmetric functions and therefore each sum bit s_i can be calculated with a periodic symmetric function $F_{s,i}(X)$ that has a period of $T = 2^{i+1}$. Remind that each input bit a_k, b_k has a weight $w_k = 2^k$. Thus each input bit with $k > i$ contributes a whole number of periods to X and has therefore no effect on the value of $F_{s,i}(X)$. Each sum bit s_i can therefore be calculated with a periodic symmetric function $F_{s,i}$ as:

$$s_i = F_{s,i}(\sum_{k=0}^{i} 2^k(a_k + b_k)). \tag{5.1}$$

In the same way many other arithmetic functions can be described by generalized periodic symmetric functions. In the next section a building block is proposed that performs a generalized periodic symmetric function.

5.2 Multiple-Input PSF Building Block

In Section 2.2.2 the PSF building block, that computes the periodic symmetric function over an analog input, was introduced. In order to implement a generalized periodic symmetric function, a building block is needed that evaluates the periodic symmetric function on multiple Boolean inputs, and that allows each input to have a different weight. Since the PSF block, as depicted in Figure 2.8, has only one input the block has to be extended. Remind that the PSF block is composed out of two stages, a SET electron trap on the input and a static inverting buffer on the output.

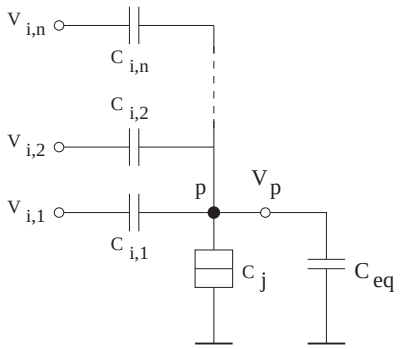


Figure 5.2: Multiple-input SET electron trap.

The topology of the electron trap can be modified to allow multiple inputs driving the electron trap in the same time. The resulting circuit topology is presented in Figure 5.2. The capacitance C_{eq} represents the equivalent capacitance of the circuit connected to the output (V_p) of the electron trap.

Every input $V_{i,x}$ contributes to the output voltage according to the following expression

$$V_{p,x} = \frac{C_{i,x}}{C_{\Sigma p}} V_{i,x}, \tag{5.2}$$

where $C_{\Sigma p} = \sum_{x=1}^{n} C_{i,x} + C_j + C_{eq}$ is the total capacitance connected to node p. The total output voltage is the sum of the contributions of all inputs:

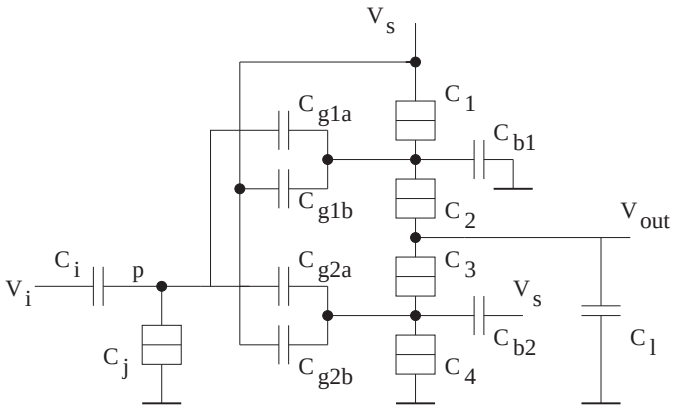
$$V_p = \sum_{x=0}^{n} V_{p,x}. \tag{5.3}$$

From Equation (5.2) it can be observed that every input $V_{i,x}$ is contributing to the voltage on node p according to the size of capacitor $C_{i,x}$. Thus by choosing different values for $C_{i,x}$, inputs can be given different weights. Let $V_{i,high}$ be the input voltage representing logic '1' and let w_x be the weight of input x, the corresponding capacitance of input x can be evaluated as:

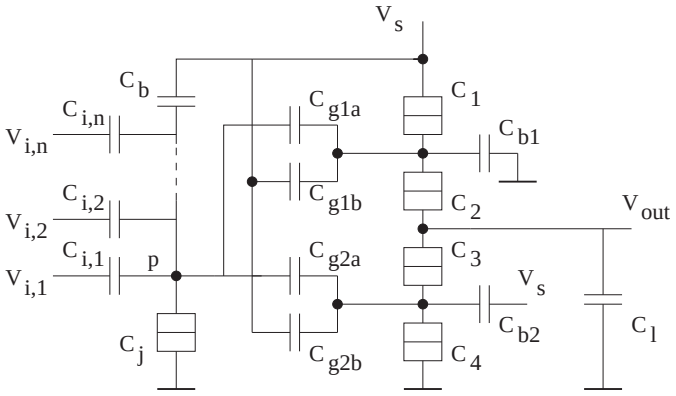
$$C_{i,x} = \frac{q_e}{2 w_x V_{i,high}}. \tag{5.4}$$

Using the circuit topology of the multiple-input electron trap of Figure 5.2, a multiple-input PSF block can be build. The new multiple-input PSF block is presented in Figure 5.3(b), while the old single-input PSF block is depicted in Figure 5.3(a) for comparison.

Besides the larger number of inputs, the new topology has a second important improvement. To explain the second improvement we first take a look at the old topology, that has one analog input, which we assume to have a weight of one. The transfer function of the circuit is depicted in Figure 5.4(a). The period of the transfer function, which is determined by the value of C_i, was set to two units because the input has a weight of one. From the transfer function it is seen that the first negative transition is



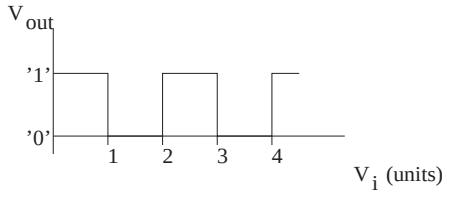
(a) Single-input PSF.



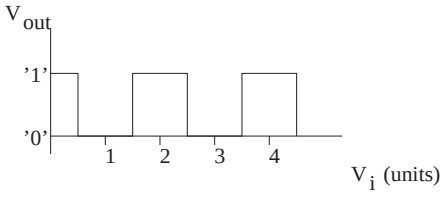
(b) Multiple-input PSF.

Figure 5.3: Old and New PSF implementation.

located at exactly an input voltage corresponding to one unit. If, due to various effects (cross-talking, impurities, parameter deviation, etc.) the input voltage is a little less than the voltage corresponding to one unit, the output of the PSF block would be logic '1' instead of the expected logic '0'.



(a) without bias.



(b) with bias.

Figure 5.4: PSF building block transfer function.

The solution to this problem is to add a bias to the voltage of node p corresponding

$C_{a0,s0} = 5aF$	$C_{a0,s1} = 2.5aF$	$C_{a0,s2} = 1.25aF$	$C_{a0,s3} = 0.625aF$
	$C_{a1,s1} = 5aF$	$C_{a1,s2} = 2.5aF$	$C_{a1,s3} = 1.25aF$
		$C_{a2,s2} = 5aF$	$C_{a2,s3} = 2.5aF$
$C_{b0,s0} = 5aF$	$C_{b0,s1} = 2.5aF$	$C_{b0,s2} = 1.25aF$	$C_{b0,s3} = 0.625aF$
	$C_{b1,s1} = 5aF$	$C_{b1,s2} = 2.5aF$	$C_{b1,s3} = 1.25aF$
		$C_{b2,s2} = 5aF$	$C_{b2,s3} = 2.5aF$

Table 5.2: Capacitance values for 3-bit PSF based adder.

to a half unit. This causes all transitions to move to the left by a half unit (see Figure 5.4(b)). The first negative transition has moved from an input of one unit to a half unit. Note that the input, though analog, is discrete in nature and only takes values of whole units. Therefore, adding a bias of a half unit results in maximal robustness of the implementation.

5.3 PSF based Addition

When adding two binary numbers $A = \{a_{n-1}, ..., a_1, a_0\}$ and $B = \{b_{n-1}, ..., b_1, b_0\}$ the result is a sum $S = \{s_n, ..., s_1, s_0\}$. As explained in Section 5.1 each sum bit s_i can be calculated with a generalized periodic symmetric function. Thus a PSF based addition scheme can be build by utilizing a multiple input PSF block for each output bit and connecting to it all the necessary inputs using the proper weights.

A 3-bit PSF based adder was build and simulated. The schematic of the circuit is depicted in Figure 5.5. We assumed a supply voltage $V_s = 16mV$ and that logic '1' is represented as $16mV$. Further we assumed that the inputs are driven by ideal voltage sources.

The 3-bit PSF adder has 6 inputs $\{a_2, a_1, a_0, b_2, b_1, b_0\}$ which have weights $\{4, 2, 1, 4, 2, 1\}$ and four outputs $\{s_3, s_2, s_1, s_0\}$ which have weights $\{8, 4, 2, 1\}$. To denote the capacitor, connecting input x to the PSF block generating output y, the notation $C_{x,y}$ is used. To calculate the values of $C_{x,y}$ Equation (5.4) was used, which resulted in the values in Table 5.2.

The values for the bias capacitances of the electron traps were also calculated using Equation (5.4) resulting in the following values: $C_{bias0} = 2.5aF$, $C_{bias1} = 1.25aF$, $C_{bias2} = 0.625aF$, $C_{bias3} = 0.313aF$. The capacitance C_j of the tunnel junction of the electron traps was set to $2.5aF$. As stated before the exact value of this capacitance is of not much importance however a good choice is a capacitance value in the same order of magnitude as the input capacitances of the electron trap. A value too large would cause the output voltage of the electron trap to decrease, which results in a higher required accuracy for the output buffer of the PSF block. The capacitance values of the output buffer of the PSF block (see Figure 5.3(b)) were calculated as follows: $C_1 = C_4 = 0.1aF$, $C_2 = C_3 = 0.5aF$, $C_{b1} = C_{b2} = 4.28aF$, $C_{g1a} = C_{g2a} = 2.72aF$, $C_{g1b} = C_{g2b} = 2.28aF$, $C_l = 9aF$.

The simulation results of the 3-bit PSF based adder are presented in Figure 5.6 and they indicate that the PSF based adder functions correctly. The adder requires 70 circuit

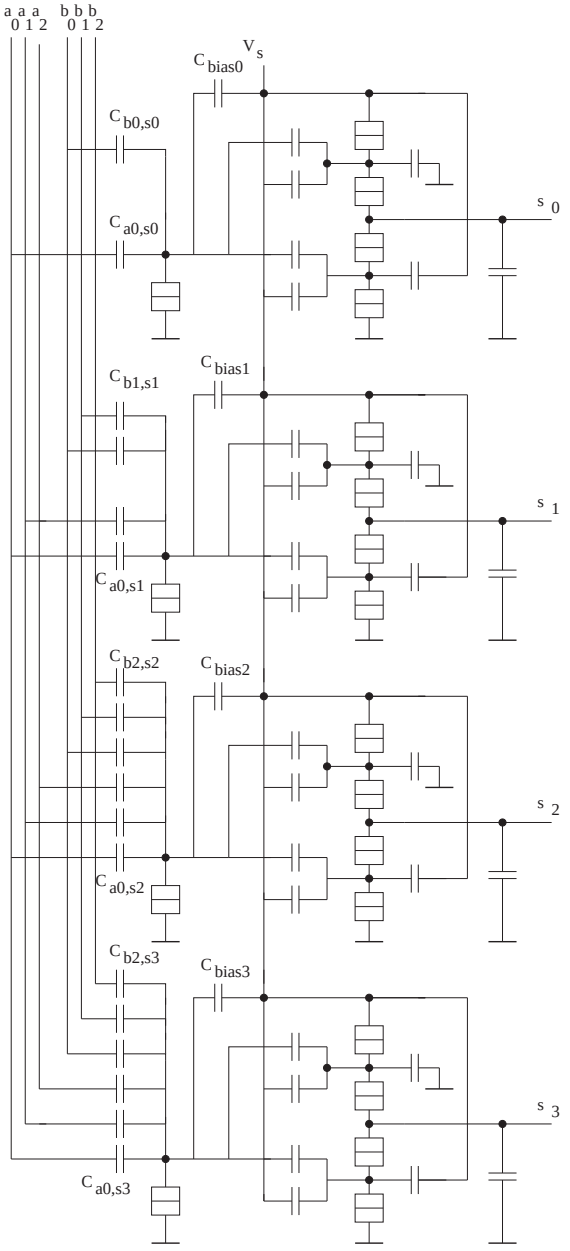


Figure 5.5: 3-bit PSF based addition scheme.

elements and, assuming an error probability $P_{error} = 10^{-8}$, has a delay of $16.0ns$. In general, for an n-bit PSF based adder $n^2 + 16n + 13$ circuit elements are needed, thus the area is in the order of $O(n^2)$. However for small n ($n \ll 16$) the area cost can be considered as being linear to the number of inputs. The delay is determined by the slowest PSF block, which is the one producing the most significant bit. The delay of the least significant PSF block is approximate $2ns$ but doubles for every next PSF block, and therefore the delay is in the order of $O(n^2)$.

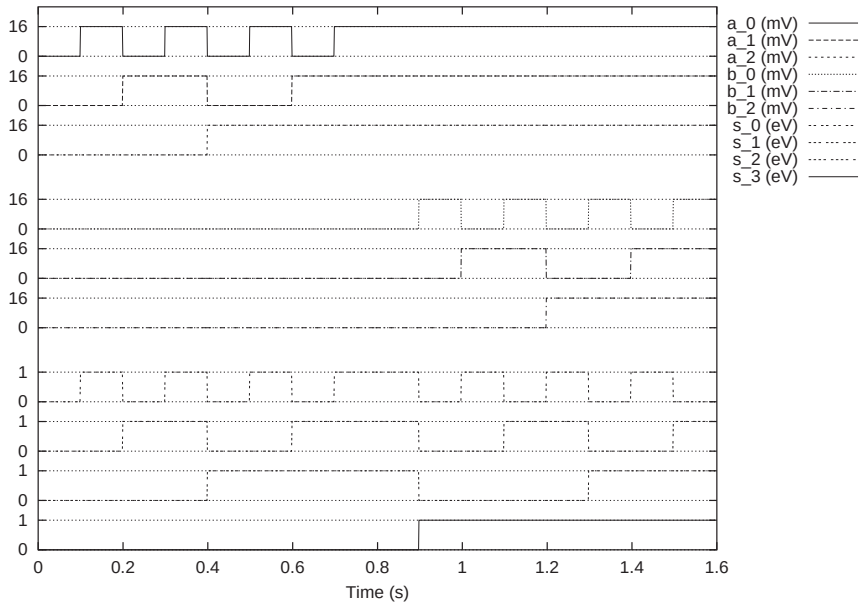
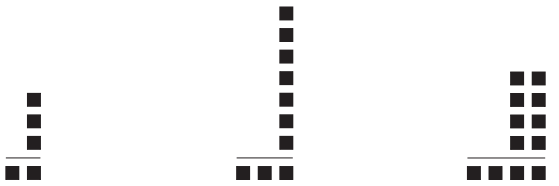


Figure 5.6: Simulation results for 3-bit PSF based adder.

5.4 PSF based Block Save Adder

Multi-operand addition is often necessary for the realization of some arithmetic operations like multiplication, especially when delay is of great importance. An often used approach to implement the addition of n m-bit unsigned numbers comprises two steps. The inputs form an mxn matrix, which in the first step is reduced to two rows. Traditionally, this step is performed by the use of counters, which avoid carry propagation and thus minimizes the delay. A counter with x inputs counts the number of ones on the input and represents the output as a y-bit number (see Figure 5.7), where $y = \lceil log_2(x+1) \rceil$. Such a counter is generally referred to as a x/y counter. The full reduction of a mxn matrix to two rows requires several stages of counters. The matrix is partitioned in horizontal slices. In every stage one or several rows of x/y counters are used, each of them reducing a slice (or column) of x bits to a row of y bits.



(a) 3/2 counter. (b) 7/3 counter. (c) BSA(4,2,4).

Figure 5.7: Schematic of counters and block save adder.

Another interesting approach is to partition the matrix in blocks and to use Block

$C_{a0,0,s0} = 5aF$	$C_{a0,0,s1} = 2.5aF$	$C_{a0,0,s2} = 1.25aF$	$C_{a0,0,s3} = 0.625aF$
	$C_{a1,0,s1} = 5aF$	$C_{a1,0,s2} = 2.5aF$	$C_{a1,0,s3} = 1.25aF$
$C_{a0,1,s0} = 5aF$	$C_{a0,1,s1} = 2.5aF$	$C_{a0,1,s2} = 1.25aF$	$C_{a0,1,s3} = 0.625aF$
	$C_{a1,1,s1} = 5aF$	$C_{a1,1,s2} = 2.5aF$	$C_{a1,1,s3} = 1.25aF$
$C_{a0,2,s0} = 5aF$	$C_{a0,2,s1} = 2.5aF$	$C_{a0,2,s2} = 1.25aF$	$C_{a0,2,s3} = 0.625aF$
	$C_{a1,2,s1} = 5aF$	$C_{a1,2,s2} = 2.5aF$	$C_{a1,2,s3} = 1.25aF$
$C_{a0,3,s0} = 5aF$	$C_{a0,3,s1} = 2.5aF$	$C_{a0,3,s2} = 1.25aF$	$C_{a0,3,s3} = 0.625aF$
	$C_{a1,3,s1} = 5aF$	$C_{a1,3,s2} = 2.5aF$	$C_{a1,3,s3} = 1.25aF$

Table 5.3: Capacitance values for BSA(4,2,4).

Save Adders (BSAs). Figure 5.7(c) depicts the schematic of a BSA(4,2,4), a block save addition of four rows and two columns producing a four bit output. Using this approach of partitioning results in a very small depth network, assuming the right block size was chosen [34].

Block save addition can easily be expressed as a generalized periodic symmetric function. Assuming a block of k columns and l rows containing the elements $a_{k,l}$, the sum bits can be described as a PSF of

$$X = \sum_{i=0}^{k} 2^i \sum_{j=0}^{l} a_{i,j}, \tag{5.5}$$

where $a_{0,0}$ is the bit in the right bottom corner. Thus a block save adder can be implemented using multiple input PSF building blocks.

Figure 5.8 depicts the PSF based implementation of a BSA(4,2,4). The multiple input PSF block generating s_0 has a period of 2 and is therefore only connected to inputs with a weight of 1 ($a_{0,j}$). To denote the capacitor, connecting input x to the PSF block generating output y, the notation $C_{x,y}$ is used. To calculate the values of $C_{x,y}$ Equation (5.4) was used, which resulted in the values in Table 5.3.

The values for the bias capacitances of the electron traps were also calculated using Equation (5.4) resulting in the following values: $C_{bias0} = 2.5aF$, $C_{bias1} = 1.25aF$, $C_{bias2} = 0.625aF$, $C_{bias3} = 0.313aF$. The capacitance C_j of the tunnel junction of the electron traps was set to $2.5aF$ for the same reasons mentioned in the previous section. The capacitance values of the output buffer of the PSF block were the same as those used in the previous section.

The simulation results for the BSA(4,2,4) based adder are presented in Figure 5.9 and they indicate that the block save adder functions correctly. The adder requires 80 circuit elements and, assuming an error probability $P_{error} = 10^{-8}$, has a delay of $16.0ns$.

5.5 SEEL Buffered PSF Addition Scheme

The example circuits presented in Section 5.3 and Section 5.4 assumed, among others, ideal input voltage sources. However, when using Single Electron Encoded Logic (SEEL) gates, logic values are represented as a net charge of one electron on a charge reservoir,

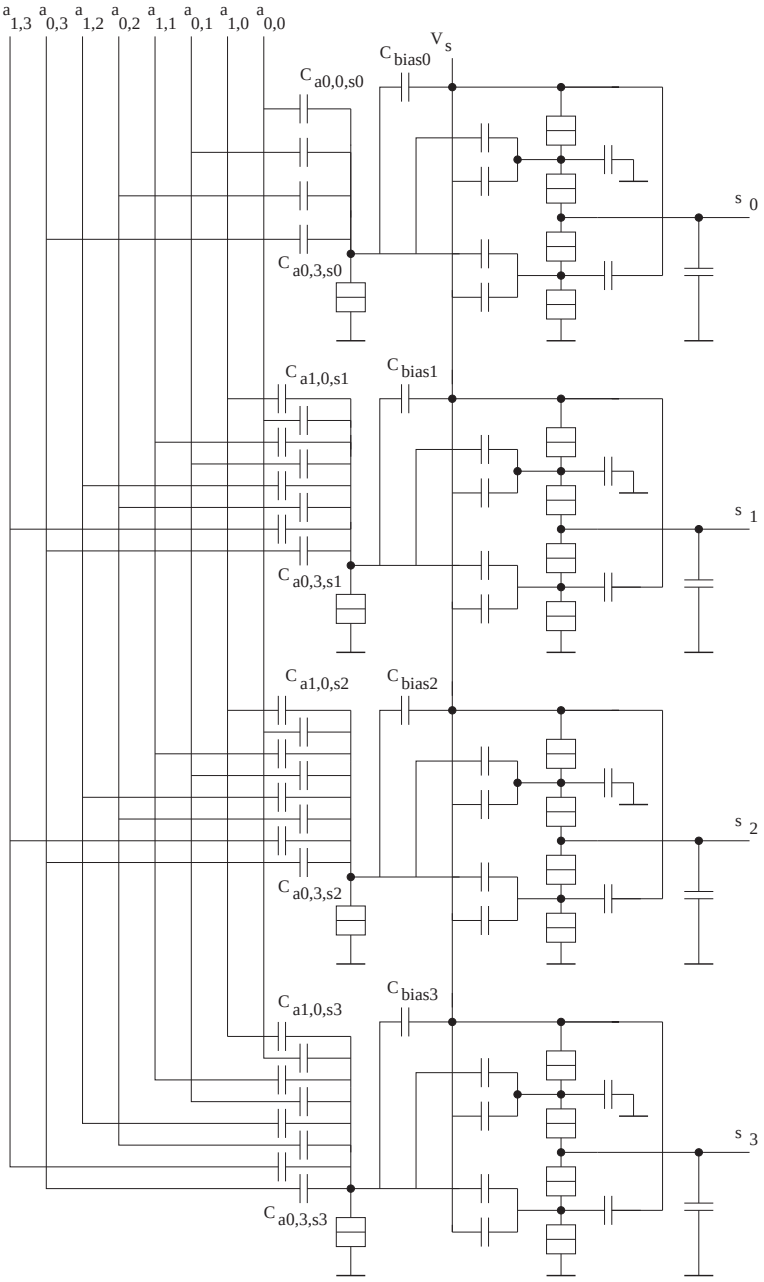


Figure 5.8: PSF based BSA(4,2,4) scheme.

which is a capacitor. Connecting blocks with relatively large input capacitances to the reservoir causes the voltage of the reservoir to decrease because of capacitive division. Thus directly connecting the output of a SEEL gate to the input of several PSF building blocks causes problems. In this section a buffered version of the 3-bit PSF based adder, that can be driven by SEEL gates, is presented. Using this buffered addition scheme a PSF based adder can be designed stand alone and can be used as a building block for

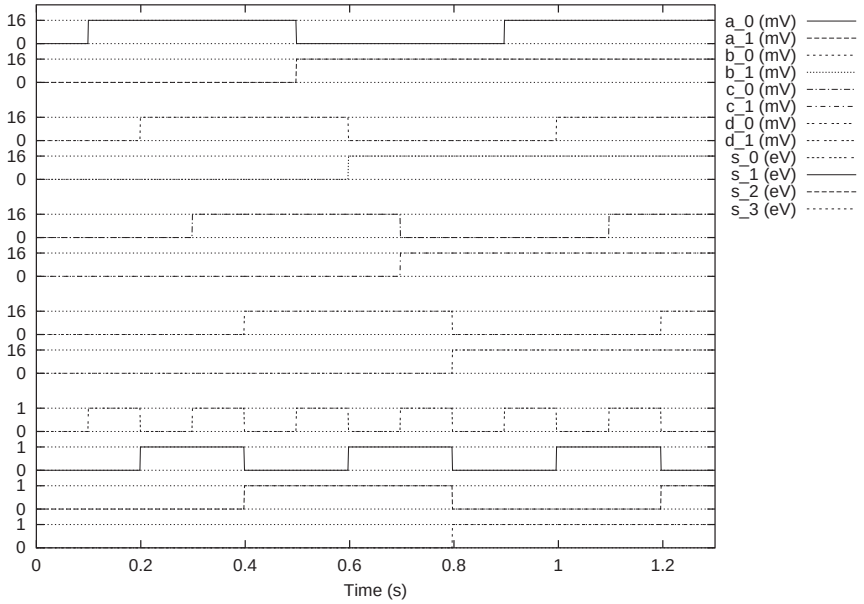


Figure 5.9: Simulation results for BSA(4,2,4).

designing larger SEEL circuits.

In this section first a mathematical model is presented that describes the connection of a static inverting buffer to a PSF building block. Second the buffered 3-bit PSF based addition scheme, which can be embedded in a SEEL environment, and the simulation results of are presented.

5.5.1 Connecting Buffer to PSF Block

The output signal of a SEEL gate is generated by a static inverting buffer [25], which is depicted in Figure 5.10. The value logic '1' is represented as a net charge of one electron on node o, which results in an output voltage V_o of approximate $16mV$ assuming $C_l = 10aF$. When connecting a building block to the output of the buffer, it is assumed that its input capacitance C_i is much smaller than the load capacitor of the buffer, that is $C_i \ll C_l$. If that is not the case, the output voltage of the buffer would decrease and it might even cause the buffer to malfunction.

This situation occurs when connecting a buffer to the input of the PSF based addition scheme, or even to a single PSF block. Assuming that logic '1' is represented as a voltage of V_1 Volt, using Equation (2.10) the input capacitances (C_i) of the non-buffered PSF based addition scheme are calculated to have values up to $\frac{q_e}{2V_1}F$. But, in order to generate a voltage of magnitude V_1 on the output of a buffer, a load capacitor $C_l = \frac{q_e}{V_1}$ is needed. This implies that $C_l = 2C_i$ and the condition that $C_i \ll C_l$ is not fulfilled. The large total capacitance on the buffer output causes the output voltage to decrease, which in turn causes a wrong result on the output of the electron trap.

The solution to this problem is an integral design of the buffer and the electron trap, which is the first stage of the PSF block. More specific, the right combination of C_l and

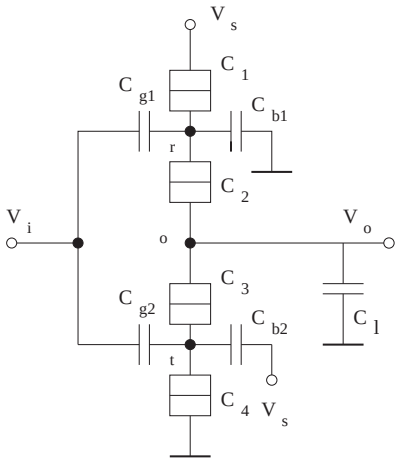


Figure 5.10: Static inverting buffer.

C_i, resulting in a correct functioning buffer and electron trap, needs to be calculated. To do so, two different models are used to describe the connection between the buffer and the electron trap. In the first model, the focus is on the output voltage of the buffer and for this purpose the electron trap is reduced to some equivalent capacitance. In the second model, the focus is on the output voltage of the electron trap and now the buffer can be reduced to some equivalent capacitor.

The first model is used to calculate the output voltage of the buffer. The topology in Figure 5.10 is used for this purpose, where the load capacitor C_l is replaced by a capacitor C_{leq} denoting the equivalent capacitance of both C_l and the electron trap. The capacitance C_{leq} is expressed as (see Figure 5.3 for the capacitance labels):

$$C_{leq} = \frac{C_i[C_j + C_{g1a} + C_{g2a}]}{C_i + C_j + C_{g1a} + C_{g2a}} + C_l. \tag{5.6}$$

The output voltage V_o of the buffer was derived as (see Figure 5.10 for the parameter labels):

$$V_o = \frac{V_s - \frac{[(C_{b1}+C_{g1})V_s - C_{g1}V_i - q_r]}{C_{\Sigma r}-C_2} + \frac{C_{\Sigma r}q_o}{C_2[C_{\Sigma r}-C_2]} + \frac{C_{\Sigma r}C_3}{C_2C_{\Sigma t}[C_{\Sigma r}-C_2]}[C_{g2}V_i + C_{b2}V_s + q_t]}{\left\{1 + \frac{C_{\Sigma r}}{C_2[C_{\Sigma r}-C_2]}\left[C_{leq} + \frac{C_3}{C_{\Sigma t}}(C_{\Sigma t} - C_3)\right]\right\}} \tag{5.7}$$

where $C_{\Sigma r}$ and $C_{\Sigma t}$ are the total capacitances connected to nodes r and t, respectively.

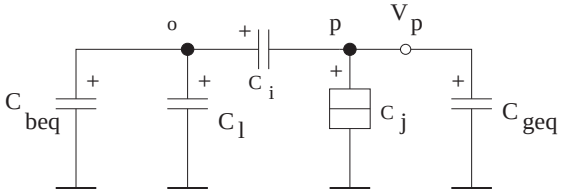


Figure 5.11: Second model of connecting buffer and electron trap.

The second model is used to calculate the output voltage of the electron trap, resulting from the output voltage of the buffer as calculated using the first model. In the second model the buffer is reduced to some equivalent capacitance C_{beq}, which can be approximated by choosing $C_{beq} = C_2 + C_3$. Also the buffer connected to the output of the electron trap was reduced to $C_{geq} = C_{g1a} + C_{g2a}$. The resulting topology of the second model is depicted in Figure 5.11.

Using the second model the equation for the output voltage of the electron trap was derived as

$$V_p = \frac{q_p + C_i V_o}{C_{\Sigma p}}, \tag{5.8}$$

where $C_{\Sigma p}$ is the total capacitance connected to node p. The factor q_p is the net charge on node p due to tunnelled electrons. For the current derivations we can safely assume $q_p = 0$. The critical voltage of the tunnel junction is derived as

$$V_c = \frac{q_e}{2\left[C_j + C_{geq} + \frac{C_i[C_{beq}+C_l]}{C_{\Sigma o}}\right]}, \tag{5.9}$$

where $C_{\Sigma o}$ is the total capacitance connected to node o. If w is the weight of the input, the critical voltage can be related to the voltage V_p by the expression $V_c = wV_p$. Substitution of the expressions from Equations (5.8) and (5.9) results in the following expression:

$$\frac{wC_i V_o}{C_{\Sigma p}} = \frac{q_e}{2\left[C_j + C_{geq} + \frac{C_i[C_{beq}+C_l]}{C_{\Sigma o}}\right]}. \tag{5.10}$$

Equations (5.6), (5.7) and (5.10) are describing the connection of the buffer and the electron trap. Solving this set of equations results in the correct value for C_l and C_i. Note that this set of equations is non-linear and to solve it a numerical method is required.

5.5.2 Buffered PSF based Addition Scheme

The mathematical model, as presented above, describes the connection of one buffer to one PSF block, but can be easily extended to describe the connection of one buffer to multiple PSF blocks. If one buffer could drive multiple PSF blocks, the PSF based addition scheme of Figure 5.5 could be made SEEL gate compatible by placing one buffer at every input. Our experiments indicate that it is not possible to have one buffer drive multiple PSF blocks and have all electron traps function correctly, as no solution to the set of equations describing that topology could be found.

Therefore the buffered PSF addition scheme, as depicted in Figure 5.12, was designed in which every input of the PSF blocks is driven by its own buffer. The static inverting buffers, of which topology is depicted in Figure 5.10, are drawn as a single element. To calculate all parameter values, one more issue had to be resolved, namely the value of the bias capacitances.

To obtain maximal robustness a bias of a magnitude of half unit was added to node p of the electron trap (see Figure 5.3(b)), which was done using capacitor C_b. Using

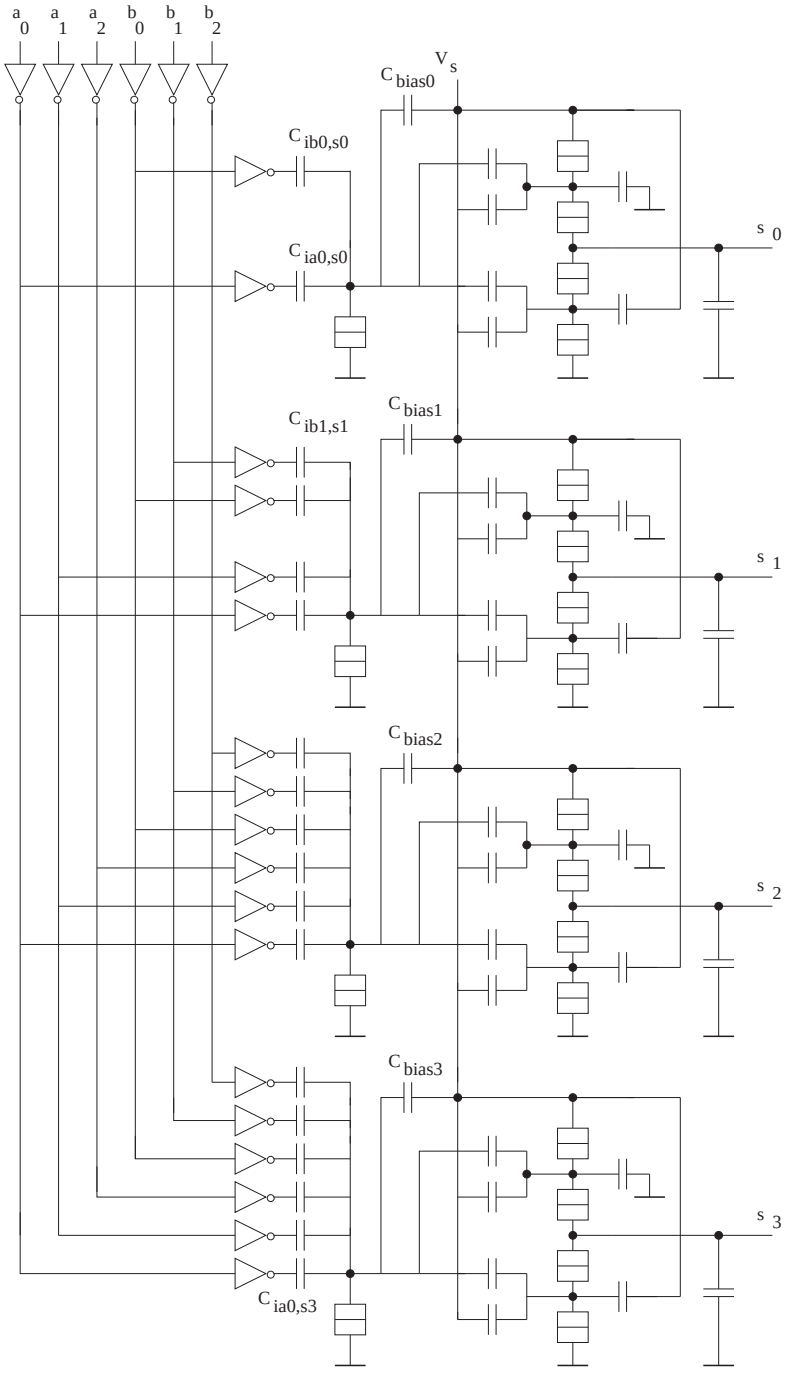
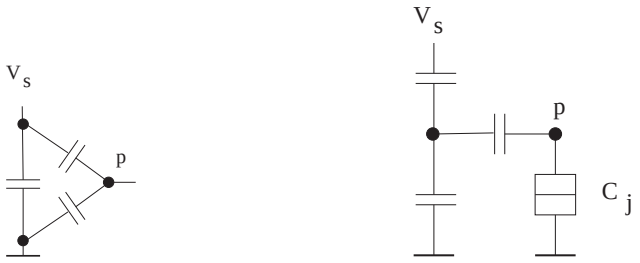


Figure 5.12: Buffered 3-bit PSF addition scheme.

static inverting buffers to drive the inputs of the electron trap, the bias on the output of these induces a bias on node p too. Since the latter one is fixed, the bias added through capacitor C_b should be changed such that the total bias on node p is the targeted bias of a magnitude of half unit.

To calculate the right value of C_b the following steps are performed. First, it is calculated what bias is induced on node p by the buffers. Second, it is determined what magnitude of bias needs to be added to the voltage of node p and what value for capacitor C_b is needed for that.



(a) model of buffer. (b) final total model.

Figure 5.13: Circuit models for calculating the bias voltage.

In order to calculate the bias voltage V_{bind} induced by the buffers, the circuits of these buffers had to be simplified. First, since the buffers are inverting, to obtain a 'low' output, the input of the buffer should be '$high$' which corresponds to $16mV$ and which is equal to the supply voltage. Therefore, the input of the buffer and the supply voltage are considered to be the same node. Second, the tunnel junctions are replaced with capacitors, maintaining the capacitance value. Third, the obtained circuit is simplified by joining together capacitances and star-triangle transformations. The resulting circuit topology is depicted in Figure 5.13(a).

So far, we only discussed the bias induced by the buffers on the input of the PSF block. But in the PSF block itself, there is also a buffer that induces some bias voltage on node p. In the previous section this bias was neglected since it is relatively small and can be neglected for most cases. In this section we do take this bias into account by simplifying the output buffer of the PSF block in the same way as it was done with the other buffers.

Once every buffer is simplified to the model of Figure 5.13(a), the resulting circuits can easily be simplified into one circuit with the same topology by adding all parallel capacitances. By performing one more triangle-star transformation, the circuit topology depicted in Figure 5.13(b) is obtained, in which the tunnel junction is displayed too. From this circuit the bias voltage on node p can be calculated.

The second step in calculating the value of C_b is to determine what bias voltage needs to be added to node p. For this purpose the bias voltage is expressed as a fraction of V_c, the critical voltage of the tunnel junction of the electron trap. The critical voltage is expressed as $V_c = \frac{q_e}{2C_{\Sigma p}}$ where $C_{\Sigma p}$ is the total capacitance connected to node p seen from the tunnel junction's perspective. This means that not only the capacitors connected directly to node p should be considered but also the circuits behind them.

The targeted bias fraction is $\frac{1}{2w}$, where w is the weight of the output of the PSF block, while the induced bias fraction is $\frac{V_{bind}}{V_c}$. If the induced bias is smaller than the targeted bias, simply the difference can be added. If the induced bias is larger than

$C_{ia0,s0} = 4.96aF$ $C_{la0,s0} = 4aF$	$C_{ia0,s1} = 2.54aF$ $C_{la0,s1} = 6.5aF$	$C_{ia0,s2} = 1.27aF$ $C_{la0,s2} = 7.8aF$	$C_{ia0,s3} = 0.626aF$ $C_{la0,s3} = 8.4aF$
	$C_{ia1,s1} = 4.96aF$ $C_{la1,s1} = 4aF$	$C_{ia1,s2} = 2.54aF$ $C_{la1,s2} = 6.5aF$	$C_{ia1,s3} = 1.27aF$ $C_{la1,s3} = 7.8aF$
		$C_{ia2,s2} = 4.96aF$ $C_{la2,s2} = 4aF$	$C_{ia2,s3} = 2.54aF$ $C_{la2,s3} = 6.5aF$
$C_{ib0,s0} = 4.96aF$ $C_{lb0,s0} = 4aF$	$C_{ib0,s1} = 2.54aF$ $C_{lb0,s1} = 6.5aF$	$C_{ib0,s2} = 1.27aF$ $C_{lb0,s2} = 7.8aF$	$C_{ib0,s3} = 0.626aF$ $C_{lb0,s3} = 8.4aF$
	$C_{ib1,s1} = 4.96aF$ $C_{lb1,s1} = 4aF$	$C_{ib1,s2} = 2.54aF$ $C_{lb1,s2} = 6.5aF$	$C_{ib1,s3} = 1.27aF$ $C_{lb1,s3} = 7.8aF$
		$C_{ib2,s2} = 4.96aF$ $C_{lb2,s2} = 4aF$	$C_{ib2,s3} = 2.54aF$ $C_{lb2,s3} = 6.5aF$

Table 5.4: Capacitance values for buffered 3-bit PSF based adder.

the targeted bias the difference should be subtracted, but the topology does not allow for that. The solution is in the periodic nature of the circuit, which allows to add a magnitude of a period to the input, without the output changing. Thus we can add a period minus the difference. In general the bias fraction r_{badd} to add is expressed as:

$$r_{badd} = mod_2 \left[\frac{1}{2w} - \frac{V_{bind}}{V_c} \right]. \tag{5.11}$$

Using Equation (5.2) the value for C_b can be calculated that adds the right bias to node p:

$$C_b = \frac{r_{badd} q_e}{2V_s}. \tag{5.12}$$

Now, the parameters of the buffered 3-bit PSF based adder were calculated as follows. The capacitances C_l and C_i of the buffers and the corresponding electron trap input were calculated in Matlab using the mathematical model as described in Subsection 5.5.1 and are stated in Table 5.4. To denote the input capacitor of the PSF block generating output y, connected to input x, the notation $C_{ix,y}$ is used. The load capacitor of the corresponding buffer is denoted as $C_{lx,y}$. The Matlab program written for this purpose is included in Appendix B.

The parameter values of the buffers were taken from [22]: $C_1 = C_4 = 0.1aF$, $C_2 = C_3 = 0.5aF$, $C_{b1} = C_{b2} = 4.25aF$, $C_{g1} = C_{g2} = 0.5aF$. For the buffers at the inputs of the adder the standard load capacitance was chosen $C_l = 9aF$. The bias capacitances were calculated using the approach described above and were calculated as $C_{bias0} = 1.75aF$, $C_{bias1} = 0.25aF$, $C_{bias2} = 9.5aF$, $C_{bias2} = 9.63aF$.

The simulation results for the buffered 3-bit PSF based adder are presented in Figure 5.14 and indicate that the buffered PSF based adder functions correctly. The total area required for the buffered adder is 286 circuit elements, which is considerably more than the non-buffered adder. On the other hand, the delay of the buffered adder is

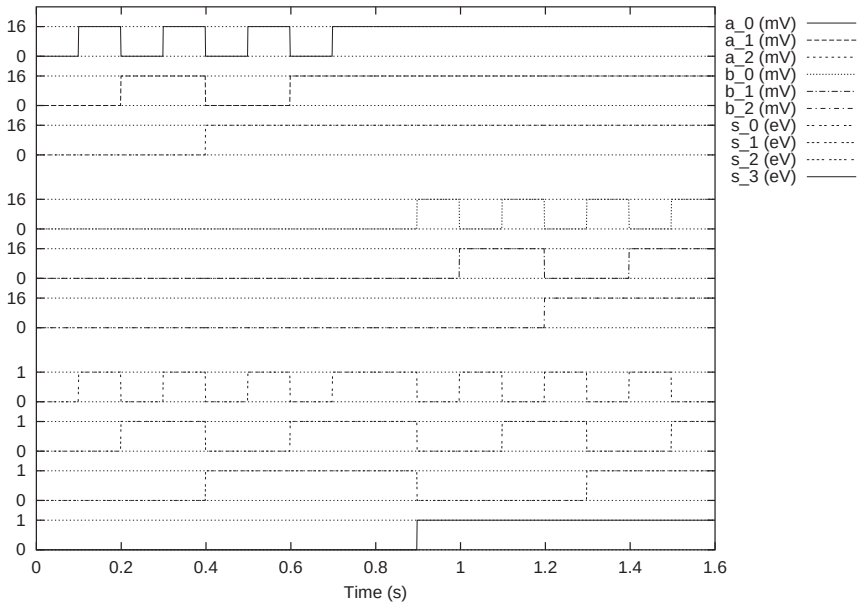


Figure 5.14: Simulation results for buffered 3-bit PSF based adder.

16.8ns, which is just slight more than the delay of the non-buffered version. In general, the required area for an n-bit buffered PSF based adder is $10n^2+61n+13$, thus the area is in the order of $O(n^2)$. However for small n ($n \ll 6$) the area cost can be considered as being linear to the number of inputs. The delay has not changed much compared to the non-buffered version and is still in the order of $O(n^2)$.

We note here that the approach we used in this section to connect the PSF based adder to SEEL gates can be used for any other PSF based scheme.

5.6 Practical Considerations

In theory any operation that can be expressed as a PSF could be implemented using the PSF based approach explained in this chapter. However, practical considerations limit the operand size for such PSF based schemes.

The first problem that arises when designing PSF based schemes with large operands, is the required accuracy. Consider the PSF block generating the most significant output bit. The number of units in a period of the electron trap, of that PSF block, is proportional to the weight of the generated output bit. Thus, the unit step size is inverse proportional to the weight. The threshold of the output buffer of the PSF block, which functions as a literal gate, is set to the middle of the appropriate unit step. With small unit steps, the margins for the threshold are small too, and therefore a high accuracy is needed. Note that the required accuracy is proportional to the number of output bits and therefore only indirectly connected to the input operand size.

The second issue is the delay of PSF based schemes, which is quadratic to the number of output bits. The delay of the PSF based addition scheme is therefore quadratic to the

number of input bits. To build an adder with large numbers of inputs, based on the PSF addition scheme, a hierarchical approach can be used. In this way the input operands are partitioned in $\frac{k}{2}$-bit blocks, where k is the maximum number of inputs a PSF adder can accommodate. For each block one PSF adder can be used and these PSF adders can be cascaded in a ripple-carry scheme or used in more efficient structures, e.g., carry look-ahead, carry-skip, etc.

6 Conclusions

In this thesis we considered arithmetic operations in Single Electron Tunneling (SET) technology and addressed three issues. Our overall investigation and achievements can be summarized as follows.

In Chapter 3 we investigated the design process of Single Electron Tunneling (SET) based building blocks in which logic values are represented by single or few electrons. A design methodology was proposed that, given a circuit topology and the corresponding targeted behaviour, allows for the derivation of the circuit parameters in an analytical way. The methodology is based on the mathematical description of the tunnel junctions in the circuit, called the characteristic equations and allows for a time effective design of SET based building blocks. Moreover the method allows for the adaptation of such blocks for the utilization in larger SET circuits. The methodology we proposed can be used for circuits operating under the Single Electron Encoded Logic (SEEL) paradigm and it is very useful as the design of SET circuits is mostly manually done, because there are no tools and only few simulators are available.

In Chapter 4 we investigated the implementation of conditional charge transport in the context of the electron counting paradigm. This is required in order to implement high radix addition schemes. A previous proposal of such a high radix addition scheme assumed the presence of a conditional charge movement ($MCke$) block which was only described as a black box. First, we proposed five possible implementations of the $MCke$ building block, each of which was described in detail and validated by means of simulation. Second, we evaluated the implementations and utilized one of them in the design of a 6-bit radix-8 adder. The resulting adder circuit was verified by means of simulation.

In Chapter 5 we investigated the implementation of periodic symmetric functions in SET technology. The subject of symmetric functions was introduced and it was explained how useful arithmetic and logic operations can be expressed as generalized periodic symmetric functions. Next, we proposed a building block, named multiple input PSF block, that can compute a generalized periodic symmetric function. Any operation that can be expressed as a generalized periodic symmetric function can be implemented using one or several of these building blocks. Two example circuits were presented, a 3-bit adder and a block save adder, which is a basic building block utilized in efficient implementation of multi operand addition. Both circuits were verified by means of simulation. We further proposed a method to embed PSF schemes in larger circuits operating according to the Single Electron Encoded Logic (SEEL) paradigm. Using that method the PSF based addition scheme was made SEEL compatible and simulation indicated that the adder functions correctly in the SEEL environment. Finally, we made some remarks on practical considerations concerning the PSF based schemes.

6.1 Proposed Research Directions

As a continuation of the research presented in this thesis we suggest the following direction for future research:

- Since only a few simulators are available, we suggest to build more advanced CAD tools. We suggest to implement the proposed design methodology in such a CAD tool. This tool could possibly also assist the designer in optimizing the circuit parameters. We also suggest to extend the current available simulators with library support to design large circuits and more advanced measurement tools for delay and energy consumption.

- In this thesis we used a rather simple and basic, though generally accepted, model of the SET phenomenon. For the purpose of this research this model was sufficient, but as SET technology matures a more detailed model will be necessary. If CAD tools are being developed we suggest that they embed additionally to the current model more accurate (lower level) ones.

- In this thesis we implemented a high radix addition scheme in the electron counting paradigm. During the master project, of which this thesis is the result, a number of high radix multiplication schemes were sketched. We suggest that these schemes are worked out in detail.

- So far, in the electron counting paradigm, only addition and multiplication have been investigated. We suggest that division and other frequently used operations are investigated. Since division is a non-linear operation we suggest that an investigation is done on which non-linear operation can be performed by SET technology. Possibly a new algorithm is needed to perform division using the available operations. Due to inherent practical limitations such an algorithm should also allow for high radix division.

- The electron counting schemes presented so far had digital inputs and digital outputs. Digital-to-analog and analog-to-digital converters were used to convert the input and output signals, while the arithmetic operations were performed on analog values. We suggest to investigate ways to avoid the continuous conversions, and instead, once working in the analog domain remain in that domain. This implies the investigation of electron counting memory, which stores analog values.

- Any computation done in SET technology is probabilistic in nature. Though the error probability can be lowered by allowing a larger delay. However, this is not an acceptable solution. Therefore we suggest that error control techniques and fault tolerant algorithms are investigated.

Bibliography

[1] http://www.lybrary.com/simon/.

[2] http://www.hana.physics.sunysb.edu/set/software/inedx.html.

[3] N. Asahi, M. Akazawa, and Y.Amemiya, *Single-Electron Logic Systems Based on the Binary Decision Diagram*, IEICE Transactions on Electronics **E81-C** (1998), no. 1, 49–56.

[4] D.V. Averin and K.K. Likharev, *Coulomb blockade of single-electron tunneling, and coherent oscillations in small tunnel junctions*, Journal of low temperature physics **62** (1986), no. 3/4, 345–373.

[5] ———, *Possible applications of the single charge tunneling*, Single Charge Tunneling - Coulomb Blockade Phenomena in Nanostructures (H. Grabert and M.H. Devoret, eds.), NATO ASI series B, vol. 294, Plenum Press and NATO Scientific Affairs Division, New York and London, 1992, pp. 311–332.

[6] D.V. Averin, L.F. Register, K.K. Likharev, and K. Hess, *Single-electron coulomb exclusion on the atomic level*, Applied Physics Letters **64** (1994), 126–128.

[7] S. Banerjee, S. Huang, and S. Oda, *Operation of Nanocrystalline-Silicon-Based Few-Electron Memory Devices in the Light of Electron Storage, Ejection and Lifetime Characteristics*, IEEE Transactions on Nanotechnology **2** (2003), no. 2, 88–92.

[8] R.H. Chen, A.N. Korotkov, and K.K. Likharev, *Single-electron Transistor Logic*, Applied Physics Letters **68** (1996), no. 14, 1954–1956.

[9] S. D. Cotofana, C. R. Lageweg, and S. Vassiliadis, *On computing addition related arithmetic operations via controlled transport of charge*, proceedings of 16th IEEE Symposium on Computer Arithmetic, June 2003, pp. 245–252.

[10] S.D. Cotofana, C.R. Lageweg, and S. Vassiliadis, *Addition Related Arithmetic Operations via Controlled Transport of Charge*, IEEE Transactions of Computers **54** (2005), no. 3, 243–256.

[11] S. Diner, D. Fargue, G. Lochak, and F. Selleri, *The wave-particle dualism - a tribute to Louis de Broglie on his 90th birthday*, D. Reidel Publishing Company, 1984.

[12] D. Esteve, *Transferring electrons one by one*, Single Charge Tunneling - Coulomb Blockade Phenomena in Nanostructures (H. Grabert and M.H. Devoret, eds.), NATO ASI series B, vol. 294, Plenum Press and NATO Scientific Affairs Division, New York and London, 1992, pp. 109–137.

[13] N. Garcia and A. Damask, *Physics for computer science students*, Springer Verlag, 1991.

[14] L.J. Geerligs, V.F. Anderegg, P.A.M. Holweg, and J.E. Mooij, *Frequency-locked turnstile device for single electrons*, Physical review letters **64** (1990), no. 22, 2691–2694.

[15] L.J. Geerligs, V.F. Anderegg, C.A. van der Jeugd, J. Romijn, and J.E. Mooij, *Influence of dissipation on the coulomb blockade in small tunnel junctions*, Europhysics Letters **10(1)** (1989), 79–85.

[16] H. Grabert and M.H. Devoret (eds.), *Single Charge Tunneling Coulomb Blockade Phenomena in Nanostructures*, NATO ASI series B, vol. 294, Plenum Press, New York, 1992.

[17] H. Grabert, G.L. Ingold, M.H. Devoret, D. Esteve, H. Pothier, and C. Urbina, *Single Electron Tunneling Rates in Multijunction Circuits*, Zeitschrift fur Physics B, Condensed Matter **84** (1991), 143–155.

[18] L. Guo, E. Leobandung, and S. Chou, *A Silicon Single-Electron Transistor Memory Operating at Room Temperature*, Science **275** (1997), 649–651.

[19] H. Iwamura, M. Akazawa, and Y. Amemiya, *Single-Electron Majority logic Circuits*, IEICE Transactions on Electronics **E81-C** (1998), 42–48.

[20] A. Korotkov and K. Likharev, *Single-Electron-Parametron-Based Logic Devices*, Journal of Applied Physics **84** (1998), no. 11, 6114–6126.

[21] A.N. Korotkov, R.H. Chen, and K.K. Likharev, *Possible performace of capacitively coupled single electron transistors in digital circuits*, Journal of Applied Physics **78** (1995), no. 4, 2520–2530.

[22] C.R. Lageweg, *Single electron tunneling based arithmetic computation*, Ph.D. thesis, TU Delft, 2004.

[23] C.R. Lageweg, S.D. Cotofana, and S. Vassiliadis, *A Linear Threshold Gate Implementation in Single Electron Technology*, IEEE Computer Society Workshop on VLSI, April 2001, pp. 93–98.

[24] ———, *Single Electron Encoded Logic Circuits*, Semiconductor Advance for Future Electronics (SAFE), November 2001.

[25] ———, *Static buffered set based logic gates*, 2nd IEEE Conference on Nanotechnology (NANO), August 2002, pp. 491–494.

[26] K.K. Likharev, *Single-Electron Transistors: Electronic Analogs of the DC Squids*, IEEE Transactions on Magnetics **MG-23** (1987), 1142–1145.

[27] ———, *Correlated Discrete Transfer of Single Electrons in Ultrasmall Tunnel Junctions*, IBM Journal of Research and Development **32** (1988), no. 1, 144–158.

[28] ———, *Single-Electron Devices and Their Applications*, Proceeding of the IEEE **87** (1999), no. 4, 606–632.

[29] K.K. Likharev and A.N. Korotkov, *Ultradense Hybrid SET/FET Dynamic RAM: Feasibility of Background Charge Independent Room Temperature Single Electron Digital Circuits*, Proceedings of the International Semiconductor Device Research Symposium (Charlottesville, Virginia), October 1995, pp. 355–359.

[30] K.K. Likharev and A.B. Zorin, *Theory of the bloch-wave escillations in small josephson junctions*, Journal of Low Temperature Physics **59** (1984), 347–382.

[31] T. Oya, T.Asai, T. Asai, T. Fukui, and Y. Amemiya, *A Majority-Logic Device Using an Irreversible Single-Electron Box*, IEEE Transactions on Nanotechnology **2** (2003), no. 1, 15–22.

[32] *International Technology Roadmap for Semiconductors, 2003 Edition, Executive Summary*, Downloadable from website http://public.itrs.net/Home.htm, 2003, Available from SEMATECH, ITRS department, 2706 Montoppolis Drive, Austin TX 78741, USA.

[33] J.R. Tucker, *Complementary Digital Logic based on the "Coulomb Blockade"*, Journal of Applied Physics **72** (1992), no. 9, 4399–4413.

[34] S. Vassiliadis, S.D. Cotofana, and J. Hoekstra, *Block save addition with threshold gates*, Tech. report, Laboratory of Computer Archituecture and Digital Techniques, Delft University of Technology, 1995.

[35] Rainer Waser (ed.), *Nanoelectronics and information technology - advanced electronic materials and novel devices*, 1st ed., Wiley-VCH, Berlin, 2003.

[36] C. Wasshuber, *About single-electron devices and circuits*, Ph.D. thesis, TU Vienna, 1998.

[37] W. Zwerger and M. Scharpf, *Crossover from coulomb-blockade to ohmic conduction in small tunnel junctions*, Zeitschrift für Physik B - Condensed Matter (1991), 85:421–426.

Mathematical Description of Basic SET Structures

A

This appendix gives an overview of the basic set topologies used throughout this thesis and provides the most important equations describing these structures. Included are:

A.1 $MVke$ Building Block

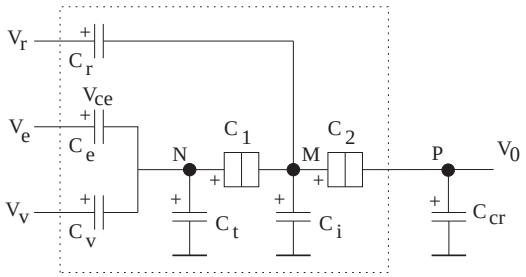


Figure A.1: $MVke$ building block implementation.

Short Notations

$$C_{\Sigma n} = C_v + C_e + C_t$$

$$C_{\Sigma m} = C_1 + C_2 + C_r + C_i$$

$$r_\alpha = \frac{C_2 + C_{cr}}{C_2 C_{cr}}[C_i + C_r] + 1$$

$$r_\beta = \frac{C_2 + C_{cr}}{C_2 C_{cr}}[C_i + C_r + C_1] + 1$$

Characteristic Equations

$$V_1 = \frac{r_\alpha \frac{C_e}{C_{\Sigma n}} V_e - \frac{C_2+C_{cr}}{C_2 C_{cr}} C_r V_r + \frac{r_\alpha}{C_{\Sigma n}} C_v V_v - \frac{q_p}{C_{cr}} - \frac{C_2+C_{cr}}{C_2 C_{cr}} q_m + \frac{r_\alpha}{C_{\Sigma n}} q_n}{r_\beta + \frac{r_\alpha C_1}{C_{\Sigma n}}}$$

$$V_2 = \frac{[C_e V_e + C_v V_v + q_n]C_1 + [C_{\Sigma n} + C_1][q_m + C_r V_r] - \{[C_{\Sigma n} + C_1][C_i + C_r] + C_{\Sigma n} C_1\} \frac{q_p}{C_{cr}}}{[C_{\Sigma n} + C_1]\left\{[C_{\Sigma m} - C_1] + \frac{C_2}{C_{cr}}[C_i + C_r]\right\} + C_{\Sigma n} C_1 \left[1 + \frac{C_2}{C_{cr}}\right]}$$

Assuming $C_2 \ll C_{cr}$ and $C_1 \ll C_{\Sigma n}$ the characteristic equations can be simplified, which results in the following:

$$V_1 = \frac{\frac{[C_{\Sigma m} - C_1]}{C_{\Sigma n}}[C_e V_e + C_v V_v + q_n] - C_r V_r - \frac{C_2}{C_{cr}} q_p - q_m}{C_{\Sigma m}}$$

$$V_2 = \frac{[C_e V_e + C_v V_v + q_n]\frac{C_1}{C_{\Sigma n}} + q_m + C_r V_r - [C_i + C_r + C_1]\frac{q_p}{C_{cr}}}{C_{\Sigma m}}$$

Critical Voltages

$$V_{c1} = \frac{q_e[C_{\Sigma n} + C_r + C_i + \frac{C_2 C_{cr}}{C_2 + C_{cr}}]}{2\left[[C_1 + C_{\Sigma n}]\left[C_r + C_i + C_1 + \frac{C_2 C_{cr}}{C_2 + C_{cr}}\right] - C_1^2\right]}$$

$$V_{c2} = \frac{q_e[C_{cr} + C_r + C_i + \frac{C_1 C_{\Sigma n}}{C_1 + C_{\Sigma n}}]}{2\left[[C_2 + C_{cr}]\left[C_r + C_i + C_2 + \frac{C_1 C_{\Sigma n}}{C_1 + C_{\Sigma n}}\right] - C_2^2\right]}$$

If the conditions $C_{\Sigma m} \ll C_{cr}$ and $C_{\Sigma m} \ll C_{\Sigma n}$ are met, these equations can be simplified to:

$$V_{c1} = V_{c2} = \frac{q_e}{2C_{\Sigma m}}$$

A.2 $MMVke$ Building Block

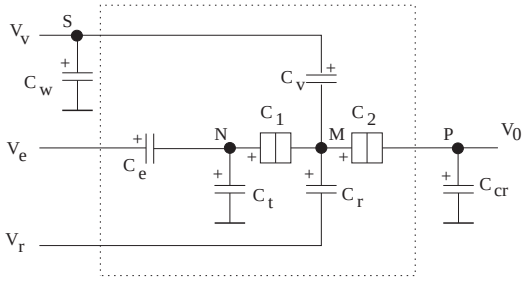


Figure A.2: $MMVke$ building block implementation.

Short Notations

$$C_{\Sigma n} = C_e + C_t$$

$$C_{\Sigma m} = C_1 + C_2 + C_r + C_\beta$$

$$C_\alpha = \frac{C_{cr} C_2}{C_{cr} + C_2}$$

$$C_\beta = \frac{C_v C_w}{C_v + C_w}$$

$$C_\gamma = \frac{C_{\Sigma n} C_1}{C_{\Sigma n} + C_1}$$

$$\alpha = 1 + \frac{C_1}{C_{\Sigma n}}$$

$$\beta = 1 + \frac{C_2}{C_{cr}}$$

Characteristic Equations

$$V_1 = \frac{[C_\alpha + C_r + C_\beta]\frac{C_e V_e + q_n}{C_{\Sigma n}} - \frac{C_\alpha}{c_{cr}} q_p - q_m - C_r V_r - \frac{C_v}{C_w + C_v} q_s}{\alpha[C_\alpha + C_r + C_\beta] + C_1}$$

$$V_2 = \frac{C_\gamma \frac{C_e V_e + q_n}{C_{\Sigma n}} - \frac{[C_\beta + C_r + C_\gamma]}{C_{cr}} q_p + q_m + C_r V_r + \frac{C_v}{C_w + C_v} q_s}{\beta[C_\beta + C_r + C_\gamma] + C_2}$$

Assuming $C_2 \ll C_{cr}$ and $C_1 \ll C_{\Sigma n}$ the characteristic equations can be simplified, which results in the following:

$$V_1 = \frac{[C_{\Sigma m} - C_1]\frac{C_e V_e + q_n}{C_{\Sigma n}} - \frac{C_2}{C_{cr}} q_p - q_m - C_r V_r - \frac{C_v}{C_w + C_v} q_s}{C_{\Sigma m}}$$

$$V_2 = \frac{C_1 \frac{C_e V_e + q_n}{C_{\Sigma n}} - \frac{[C_{\Sigma m} - C_2]}{C_{cr}} q_p + q_m + C_r V_r + \frac{C_v}{C_w + C_v} q_s}{C_{\Sigma m}}$$

Critical Voltages

$$V_{c1} = \frac{q_e[C_\beta + C_r + C_\alpha + C_{\Sigma n}]}{2\left[[C_\beta + C_r + C_\alpha][C_{\Sigma n} + C_1] + C_{\Sigma n}C_1\right]}$$

$$V_{c2} = \frac{q_e[C_\beta + C_r + C_\gamma + C_{cr}]}{2\left[[C_\beta + C_r + C_\gamma][C_{cr} + C_2] + C_{cr}C_2\right]}$$

If the conditions $C_{\Sigma m} \ll C_{cr}$ and $C_{\Sigma m} \ll C_{\Sigma n}$ are met, these equations can be simplified to:

$$V_{c1} = V_{c2} = \frac{q_e}{2C_{\Sigma m}}$$

A.3 Threshold Gate Building Block[1]

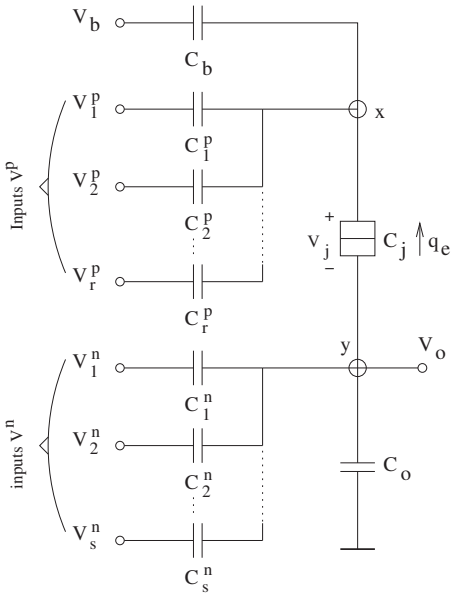


Figure A.3: n-Input linear threshold gate implementation.

Short Notations

$$C_\Sigma^p = C_b + \sum_{k=1}^{r} C_k^p$$

$$C_\Sigma^n = C_o + \sum_{l=1}^{s} C_l^n$$

$$C_\tau = C_\Sigma^p C_j + C_\Sigma^p C_\Sigma^n + C_j C_\Sigma^n$$

Characteristic Equation

$$V_j = \frac{V_b C_\Sigma^p C_\Sigma^n + C_\Sigma^n \sum_{k=1}^{r} C_k^p \left[V_k^p - V_b\right] - C_\Sigma^p \sum_{l=1}^{s} C_l^n V_l^n + C_\Sigma^n q_x - C_\Sigma^p q_y}{C_\tau}$$

Critical Voltage

$$V_c = \frac{[C_\Sigma^p + C_\Sigma^n] q_e}{2 C_\tau}$$

[1]This building block was devised by C.R. Lageweg and first published in [23]. Most equations of this section were taken from that publication.

Output Voltage

The removal of one electron from node y results in an increase of the output voltage of:

$$\delta V_o = \frac{q_e C_\Sigma^p}{C_\tau}.$$

A.4 Static Inverting Buffer Building Block [2]

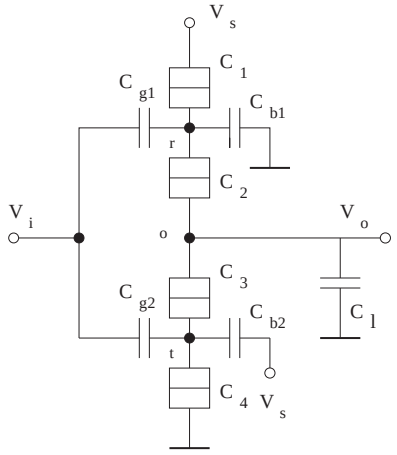


Figure A.4: Static inverting buffer implementation.

Short Notations

$$C_{\Sigma r} = C_1 + C_2 + C_{g1} + C_{b1}$$

$$C_{\Sigma 0} = C_2 + C_3 + C_l$$

$$C_{\Sigma t} = C_3 + C_4 + C_{g2} + C_{b2}$$

$$C_\alpha = \frac{C_3 C_l}{C_3 + C_l}$$

$$C_{\Sigma\alpha} = C_\alpha + C_4 + C_{g2} + C_{b2}$$

$$r_\alpha = \frac{C_3 - C_\alpha}{C_3}$$

$$C_\beta = \frac{1}{\frac{1}{C_2} + \frac{1}{C_3 + C_l} + \frac{r_\alpha C_\alpha}{C_{\Sigma\alpha} C_l}}$$

$$r_\beta = \frac{C_2 - C_1}{C_2}$$

$$C_\gamma = \frac{C_2 C_l}{C_2 + C_l}$$

$$C_{\Sigma\beta} = C_1 + C_\gamma + C_{b1} + C_{g1}$$

[2] This building block was devised by C.R. Lageweg and first published in [25]. The equations presented in this section are derived by the author of this thesis.

$$r_\gamma = \frac{C_2 - C_\gamma}{C_2}$$

$$C_\delta = \frac{1}{\frac{1}{C_3} + \frac{C_\gamma}{C_2 C_l} + \frac{R_\gamma C_\gamma}{C_{\Sigma\beta} C_l}}$$

$$C_{eq,1} = \frac{\left[C_{g2} + C_{b2} + C_4\right] C_3}{C_{\Sigma t}}$$

$$C_{eq,2} = \frac{\left[C_{g1} + C_{b1} + C_1\right] C_2}{C_{\Sigma r}}$$

Characteristic Equations

$$V_1 = \frac{V_s\left[1 + \frac{C_{g1}+C_{b1}}{C_\beta} - \frac{r_\alpha C_{b2}}{C_{\Sigma\alpha}}\right] - \frac{q_r}{C_\beta} - \frac{q_0}{C_{\Sigma o}-C_2} - \frac{r_\alpha}{C_{\Sigma\alpha}}\left[q_t + \frac{q_0 C_\alpha}{C_l}\right] - V_i\left[\frac{C_{g1}}{C_\beta} + \frac{r_\alpha C_{g2}}{C_{\Sigma\alpha}}\right]}{\frac{C_\beta + C_{\Sigma r} - C_2}{C_\beta}}$$

$$V_2 = \frac{V_s\left[1 - \frac{C_{b1}+C_{g1}}{C_{\Sigma r}-C_2} - \frac{C_\alpha C_{g2}}{C_l C_{\Sigma\alpha}}\right] + \frac{q_r}{C_{\Sigma r}-C_2} - \frac{q_o}{C_l+C_3} - V_i\left[\frac{C_\alpha C_{g2}}{C_l C_{\Sigma\alpha}} - \frac{C_{g1}}{C_{\Sigma r}-C_2}\right]}{1 + \frac{C_2}{C_{\Sigma r}-C_2} + \frac{C_2}{C_l+C_3} + \frac{C_\alpha^2 C_2}{C_l^2 C_{\Sigma\alpha}}} - \frac{\frac{C_\alpha}{C_l C_{\Sigma\alpha}}\left[q_t + \frac{C_\alpha q_o}{C_l}\right]}{1 + \frac{C_2}{C_{\Sigma r}-C_2} + \frac{C_2}{C_l+C_3} + \frac{C_\alpha^2 C_2}{C_l^2 C_{\Sigma\alpha}}}$$

$$V_3 = \frac{V_s\left[1 - \frac{C_\gamma}{C}_2 - \frac{C_{b2}}{C_{\Sigma t}-C_3} - \frac{C_\gamma}{C_l C_{\Sigma\beta}}\left(C_{b1} + C_\gamma + C_{g1}\right)\right] - V_i\left[\frac{C_{g2}}{C_{\Sigma t}-C_3} - \frac{C_\gamma C_{g1}}{C_l C_{\Sigma\beta}}\right]}{1 + \frac{C_3}{C_{\Sigma t}-C_3} + \frac{C_3}{C_l+C_2} + \frac{C_\gamma^2 C_3}{C_l^2 C_{\Sigma\beta}}} + \frac{q_o\left[\frac{1}{C_l+C_2} + \frac{C_\gamma^2}{C_l^2 C_{\Sigma\beta}}\right] + \frac{q_r C_\gamma}{C_l C_{\Sigma\beta}} - \frac{q_t}{C_{\Sigma t}-C_3}}{1 + \frac{C_3}{C_{\Sigma t}-C_3} + \frac{C_3}{C_l+C_2} + \frac{C_\gamma^2 C_3}{C_l^2 C_{\Sigma\beta}}}$$

$$V_4 = \frac{V_s\left[1 + \frac{C_{b2}}{C_\delta} - \frac{C_\gamma}{C_2} - \frac{r_\gamma}{C_{\Sigma\beta}}\left(C_{b1} + C_\gamma + C_{g1}\right)\right] + V_i\left[\frac{C_{g2}}{C_\delta} + \frac{r_\gamma C_{g1}}{C_{\Sigma\beta}}\right] + \frac{q_t}{C_\delta}}{\frac{C_\delta + C_{\Sigma t} - C_3}{C_\delta}} + \frac{q_o\left[\frac{C_\gamma}{C_2 C_l} + \frac{r_\gamma C_\gamma}{C_{\Sigma\beta} C_l}\right] + \frac{r_\gamma q_r}{C_{\Sigma\beta}}}{\frac{C_\delta + C_{\Sigma t} - C_3}{C_\delta}}$$

Critical Voltage

$$V_{c1} = \frac{q_e}{2\left[\frac{(C_{eq,1}+C_l)C_2}{C_{eq,1}+C_l+C_2} + C_{g1} + C_{b1} + C_{j1}\right]}$$

$$V_{c2} = \frac{q_e}{2\left[\frac{(C_{eq,1}+C_l)(C_1+C_{g1}+C_{b1})}{C_{eq,1}+C_l+C_1+C_{g1}+C_{b1}} + C_{j2}\right]}$$

$$V_{c3} = \frac{q_e}{2\left[\frac{(C_{eq,2}+C_l)(C_{b2}+C_{g2}+C_4)}{C_{eq,2}+C_l+C_{b2}+C_{g2}+C_4} + C_3\right]}$$

$$V_{c4} = \frac{q_e}{2\left[\frac{C_3(C_{eq,2}+C_l)}{C_3+C_{eq,2}+C_l} + C_{g2} + C_{b2} + C_{j4}\right]}$$

Output Voltage

$$V_o = \frac{V_s - \frac{(C_{b1}+C_{g1})V_s - C_{g1}V_i - q_r}{C_{\Sigma r}-C_2} + \frac{q_o C_{\Sigma r}}{C_2(C_{\Sigma r}-C_2)} + \frac{C_{\Sigma r}C_3(C_{g2}V_i + C_{b2}V_s + q_t)}{C_2 C_{\Sigma t}(C_{\Sigma r}-C_2)}}{1 + \frac{C_{\Sigma r}\left[C_l + \frac{C_3(C_{\Sigma t}-C_3)}{C_{\Sigma t}}\right]}{C_2(C_{\Sigma r}-C_2)}}$$

Source Code of the Matlab Solver

B

In Section 5.5.1 a mathematical model was presented describing the connection between a static inverting buffer and a PSF building block. Since this set of equations is non-linear it was solved by a numerical method using Matlab. This appendix presents the source code of the Matlab program that was written to solve the set of equations for the connection of one buffer to one PSF block. The function `solve_gate_psf(Ci)` contains the mathematical model and is used in conjunction with `fsolve`. The command

```
Ci=fsolve('solve_gate_psf', 10e-18, optimset('TolX',1e-22, 'TolFun',1e-8));
```

assigns the right value to `Ci` assuming the values for `k` and `Cl` were set correctly.

```
function F = solve_gate_psf(Ci);

%created by Cor Meenderinck

% this function is used to solve the nonlineair set of equations of
% the SET gate-psf connection.
% input is the value for Ci
% output is a voltage difference, that is zero if
% the right value for Ci is supplied
% this function is used in combination with fsolve:
% Ci=fsolve('solve_gate_psf', 10e-18, optimset('TolX',1e-22, 'TolFun',1e-8))
% and computes the value for Ci, given a value k and Cl.

%check usage
error (nargchk(1,1,nargin));


% start function

%adjust these values:
k=1;        % k is the weight of the PSF block
Cl=4e-18;   % Cl is the load capacitor of the gate driving the PSF block
            % choose this value such that the sum of Ci and Cl is about 10 aF

%constant values:
Vs=16e-3;
qe=1.602e-19;
```

```
%variable values:
    %gate:
Cj1=1e-19;
Cj2=5e-19;
Cj3=5e-19;
Cj4=1e-19;
Cg1=5e-19;
Cg2=5e-19;
Cb1=4.25e-18;
Cb2=4.24e-18;
Cbeq=1e-18;

    %psf:
Cj=2.5e-18;
Cg=2.5e-19;

%help variables:
Csigr=Cj1+Cj2+Cg1+Cb1;
Csigt=Cj3+Cj4+Cb2+Cg2;
a=1+Csigr/(Cj2*(Csigr-Cj2))*(Cj3/Csigt*(Csigt-Cj3));
ralpha=Csigr/(Cj2*(Csigr-Cj2));
Cle=Ci*(Cj+2*Cg)/(Ci+Cj+2*Cg)+Cl;

% first calculate the output voltage of the gate when the input is zero
% (output is high)
Vi=0;
q0=qe;
V0accent=Vs-((Cb1+Cg1)*Vs-Cg1*Vi)/(Csigr-Cj2)+Csigr*q0/(Cj2*(Csigr-Cj2))
    +Csigr*Cj3*(Cg2*Vi+Cb2*Vs)/(Cj2*Csigt*(Csigr-Cj2));
V0high=V0accent/(a+ralpha*Cle);

% next calculate the output voltage of the gate when the input is one
% (output is low, only a bias voltage is present)
Vi=16e-3;
q0=0;
V0accent=Vs-((Cb1+Cg1)*Vs-Cg1*Vi)/(Csigr-Cj2)+Csigr*q0/(Cj2*(Csigr-Cj2))
    +Csigr*Cj3*(Cg2*Vi+Cb2*Vs)/(Cj2*Csigt*(Csigr-Cj2));
V0low=V0accent/(a+ralpha*Cle);

% calculate the voltage swing on node p of electron trap produced by one gate
deltaVp=Ci*(V0high-V0low)/(Ci+Cj+2*Cg);

%calculate the critical voltage of the junction of the electron trap
Vc= qe/(2*(Cj+2*Cg+Ci*(Cbeq+Cl)/(Cl+Cbeq+Ci)));
```

```
% calculate the difference between the critical voltage and
% k times the voltage swing.
% The value for Ci is correct if the difference is zero.
F=Vc-k*deltaVp;
```

List of Publications

Accepted papers

1. C. Meenderinck, S. Cotofana, and C. Lageweg, "High Radix Addition Via Conditional Charge Transport in Single Electron Tunneling Technology", in *Proceedings of the* 16^{th} *IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, (Samos, Greece) July 2005.

2. C. Meenderinck, C. Lageweg, and S. Cotofana, "Design Methodology for Single Electron Based Building Blocks", in *Proceedings of the* 5^{th} *IEEE Conference on Nanotechnology (NANO)*, (Nagoya, Japan) July 2005.

Submitted papers

1. C. Meenderinck, and S. Cotofana, "Computing Periodic Symmetric Functions in Single Electron Tunneling Technology", in *Proceedings of the* 28^{th} *International Semiconductor Conference (CAS)*, (Sinaia, Romania), October 2005.

2. C. Meenderinck, and S. Cotofana, "Periodic Symmetric Functions and Addition Related Arithmetic Operations in Single Electron Tunneling Technology", in *Proceedings of the* 16^{th} *Annual Workshop on Circuit, Systems, and Signal Processing (ProRISC)* , (Veldhoven, Netherlands), November 2005.

Curriculum Vitae

C.H. Meenderinck was born in Amsterdam, The Netherlands, on the 29^{th} of January 1978. From 1990 he took the secondary education at the "Herbert Vissers" secondary school in Nieuw Vennep, where he graduated in 1996. In the same year he became a student of the Faculty of Electronic Engineering, Delft University of Technology, the Netherlands. After receiving his Bachelor of Science degree, he joined the Computer Engineering laboratory, led by professor Stamatis Vassiliadis, to start his MSc graduation project under the supervision of associate professor Sorin Coţofană. During his MSc graduation project he submitted three papers of which, at time of writing, two have been accepted. His research interests include computer arithmetic, nano electronics, logic design, computer aided design, and computer architecture.