SAND REPORT

SAND2004-5275
Unlimited Release
Printed October 2004

Photonic Encryption:

Modeling and Functional Analysis of All-Optical Logic

Jason D. Tang
Perry J. Robertson
Richard C. Schroeppel

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185

Sandia is a multiprogram laboratory operated by Sandia Corporation, a
Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2004-5275
Unlimited Release
Printed October 2004

Photonic Encryption:

Modeling and Functional Analysis of All Optical Logic

Jason Tang
Advanced Network Integration Department

Perry J. Robertson
RF and Opto Microsystems

Richard C. Schroeppel
Crypto and Info Systems Surety

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0806

Abstract

With the build-out of large transport networks utilizing optical technologies, more and more capacity is being made available. Innovations in Dense Wave Division Multiplexing (DWDM) and the elimination of optical-electrical-optical conversions have brought on advances in communication speeds as we move into 10 Gigabit Ethernet and above. Of course, there is a need to encrypt data on these optical links as the data traverses public and private network backbones. Unfortunately, as the communications infrastructure becomes increasingly optical, advances in encryption (done electronically) have failed to keep up. This project examines the use of optical logic for implementing encryption in the photonic domain to achieve the requisite encryption rates.

This paper documents the innovations and advances of work first detailed in “Photonic Encryption using All Optical Logic,” [1]. A discussion of underlying concepts can be found in SAND2003-4474. In order to realize photonic encryption designs, technology developed for electrical logic circuits must be translated to the photonic regime. This paper examines S-SEED devices and how discrete logic elements can be interconnected and cascaded to form an optical circuit. Because there is no known software that can model these devices at a circuit level, the functionality of S-SEED devices in an optical circuit was modeled in PSpice. PSpice allows modeling of the macro characteristics of the devices in context of a logic element as opposed to device level computational modeling. By representing light intensity as voltage, “black box” models are generated that accurately represent the intensity response and logic levels in both technologies. By modeling the behavior at the systems level, one can incorporate systems design tools and a simulation environment to aid in the overall functional design. Each black box model takes certain parameters (reflectance, intensity, input response), and models the optical ripple and time delay characteristics. These “black box” models are interconnected and cascaded in an encrypting/scrambling algorithm based on a study of candidate encryption algorithms. Demonstration circuits show how these logic elements can be used to form NAND, NOR, and XOR functions. This paper also presents functional analysis of a serial, low gate count demonstration algorithm suitable for scrambling/encryption using S-SEED devices.

[This page left intentionally blank]

Table of Contents

Table of Figures

1. Introduction

As existing transport networks evolve into intelligent all-optical networks, end-to-end connections are beginning to look like virtual fibers. The elimination of optical-electrical-optical (OEO) conversions within network equipment allows for a vast increase in network capacity due to enabling technologies such as dense wavelength division multiplexing (DWDM). This is an important trend for Sandia and the DOE complex due to the increasing need to interconnect high performance computing and visualization platforms, often in a “network protocol agnostic” fashion. However, as signaling and switching technologies progress towards an all-optical architecture, network encryption technology, which remains in the electrical domain, fails to keep pace. Because network encryption is paramount to the Sandia/DOE mission, the lack of encryption mechanisms for all-optical networks seriously limits our ability to utilize these exciting new technologies and to reach bit rates currently not viable under electronic methods.

Over the past five years, Sandia has been developing all-optical devices that perform Boolean logic functions on optical inputs and produce optical outputs without intermediate electrical conversion. These all-optical logic gates are built upon two distinct technologies – self electro-optic effect devices (SEED) and gain competition technologies. These devices form logical building blocks suitable for a designing a photonic encryptor. However, optical logic gates are just now maturing into a state of discrete operation and have yet to be demonstrated in monolithic arrangements and present a few limitations (e.g., limited cascade depth, fanout, etc.) when applied in an encryption algorithm.

We are developing a technically feasible design for a photonic encryptor that is based on a simple, but useful encryption algorithm that can be built within the limitations of the SEED and/or gain competition devices. The encryptor will be able to process an optical data stream with known characteristics and will exhibit scaling properties to bit-rates unreachable through traditional OEO methods. By designing a set of Boolean logic elements with all optical logic, we can use them in conjunction with low gate count encrypting/scrambling algorithms to demonstrate an all-optical encryptor. Because there is no known software that can model these devices at a circuit level, we have modeled the functionality of the SEED and gain competition devices in an optical circuit in PSpice. PSpice allows us to model macro characteristics of the devices in context of a logic element as opposed to device level computational modeling.

This project has examined cryptographic algorithms in detail and determined innovative implementation approaches that can be implemented within the constraints of current optical logic gate technology. In addition, novel encryption approaches that utilize other photonic properties (e.g., dispersion, polarization, etc.) that may be modulated by these devices are also being explored.

2. Photonic Encryptor Usage

Figure 1 shows the photonic encryptor designed under this project, and how it fits into a generalized photonic network architecture.

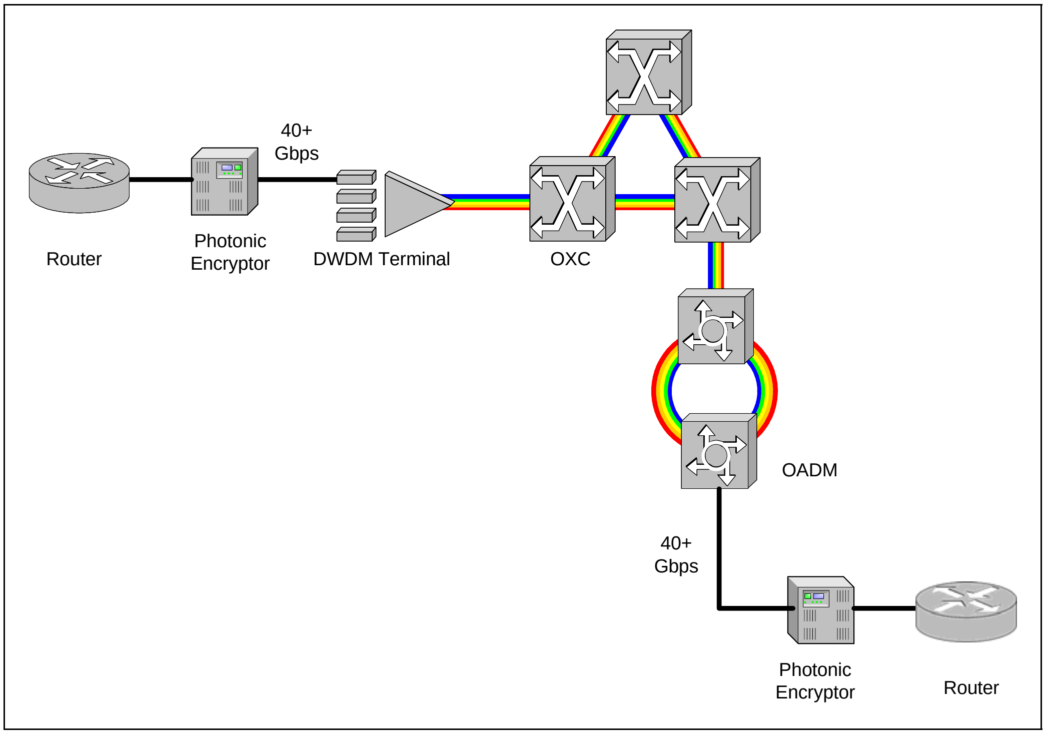


Figure 1: Photonic encryptor placement in all-optical network

In general, a photonic network may consist of the following components:

Dense Wave Division Multiplexer (DWDM)
This device takes multiple inputs with known wavelengths and framing protocols, wavelength-converts the inputs, and multiplexes the converted wavelengths onto a single fiber. These devices are typically connected to each other via a single, point-to-point connection.

Optical Add-Drop Multiplexer (OADM)
This device takes a single input with known wavelength and framing protocol, wavelength-converts it, and multiplexes the converted wavelength onto a single fiber that carries other wavelengths. In the de-multiplexing direction, this device isolates a single wavelength from a collection of wavelengths and converts it to a specified wavelength on the output interface. These devices are typically connected to each other in a ring topology.

Optical Crossconnect (OXC)
An OXC is an all-optical switch. As such, it isolates a wavelength on its input port, wavelength-converts it, and multiplexes it onto the output port. These devices can be interconnected in an arbitrary mesh.

The photonic encryptor designed in this project operates on a single wavelength with known framing protocol, as shown in Figure 1. Therefore, it is meant to connect to the input/output of a DWDM terminal or an OADM. Although techniques for broadband (multi-wavelength) encryption were considered, they were rejected for the following reasons:

Asynchronous key stream and data stream. In general, the data channels on each wavelength in a WDM network are not synchronized with each other. Therefore, a single key stream that would encrypt all wavelengths at once would be asynchronous with the data on all of the wavelengths. This is a departure from conventional cryptographic techniques, in which the key stream is synchronous with the data stream, and could present cryptanalytic challenges that this project is not prepared to accept.

Possibility for sub-rate encryption. Encryption of an arbitrarily formatted data stream with unknown bit rate could lead to a situation where the bit rate of the data stream is faster than the bit rate of the key stream. This results in re-use of key stream, which is taboo from a cryptanalytic perspective. This situation is likely for broadband encryption, as devices that can switch or operate on multiple wavelengths (e.g., micro-mirrors) are typically too slow to switch at the rates required for encryption of 40+ gigabits per second.

Wavelength constraints of photonic logic. Although other devices (e.g., micromirrors, chaotic mode-locked lasers, etc.) were considered, the speed requirements for this project called for the use of photonic logic. However, these devices operate at fixed wavelengths (usually 850 nm).

For the reasons listed above, the encryptor is designed to operate on a single wavelength with known protocol. Furthermore, the encryptor is designed to transparently pass an optical path [2]. However, the optical framing protocol that is encrypted might have overhead information (e.g., for OA&M purposes) that must bypass encryption. If so, then techniques such as those developed for optical label swapping can be used to suppress encrypted headers and substitute plaintext header information [3]. Other techniques for processing optical headers are also possible [4].

Although the single wavelength with known protocol restriction may appear to limit the encryptor's usefulness in photonic networks, it is actually a realistic configuration that would have interesting application in high speed communications. One can easily envision subscribing to a carrier's wavelength service, where the framing protocol and rate are known, but are beyond the capabilities of today's electrical domain encryption devices. By implementing photonic encryption at the point of presence, bulk encryption can be realized. Furthermore, it is expected that the techniques developed for photonic encryption will facilitate scaling of encryption data rates more readily than today's electronic implementations.

3.

PSpice Modeling

3.1. PSpice Model

Our PSpice Model is similar to the Advice model presented by Lentine [5]. Like Lentine, a fifth order polynomial has been used to characterize the optical absorption of the p.i.n diode as a function of reverse bias voltage. However, Lentine chose to characterize the device as a three terminal device, with the diode terminals (cathode and anode) being two terminals and the third being the optical input. The PSpice model presented here characterizes the SEED as a four terminal device, two electrical and two optical. From the standpoint of the simulator, all terminals are electrical. The optical inputs and outputs are simply ideal electrical terminals where 1 V electrical potential is equivalent to 1 mW optical power at the device.

3.2. Diode Electrical Model

The PSpice model is shown in Figure 2. The basic p.i.n. photodiode is modeled by a capacitor and parallel current source, G1. A lookup table is used to model the current versus voltage characteristic of the device (see Table 1). These values were taken from measured data. As can be seen in Figure 3, the piece wise linear model does a fair job of modeling the complex curve.

The diode model also contains a series resistor and series inductor whose values were derived experimentally. [6]

3.3. Responsitivity Curves

The photoelectric currents are generated by each of the four beams that shine on the device. A and B are logic inputs, C is the clock input and P is the preset input. The gain for each is light input is modeled by a linear fit where the responsivity in the discharged state (approximately -0.5 V reverse biased) is .45 A/W and in the charged state (approximately +1.5 V reverse biased) is .35 A/W. The resulting line is shown in Figure 4 and the linear fit equation is

$\mathrm{Re}\,s = .425 = .05V_r$.

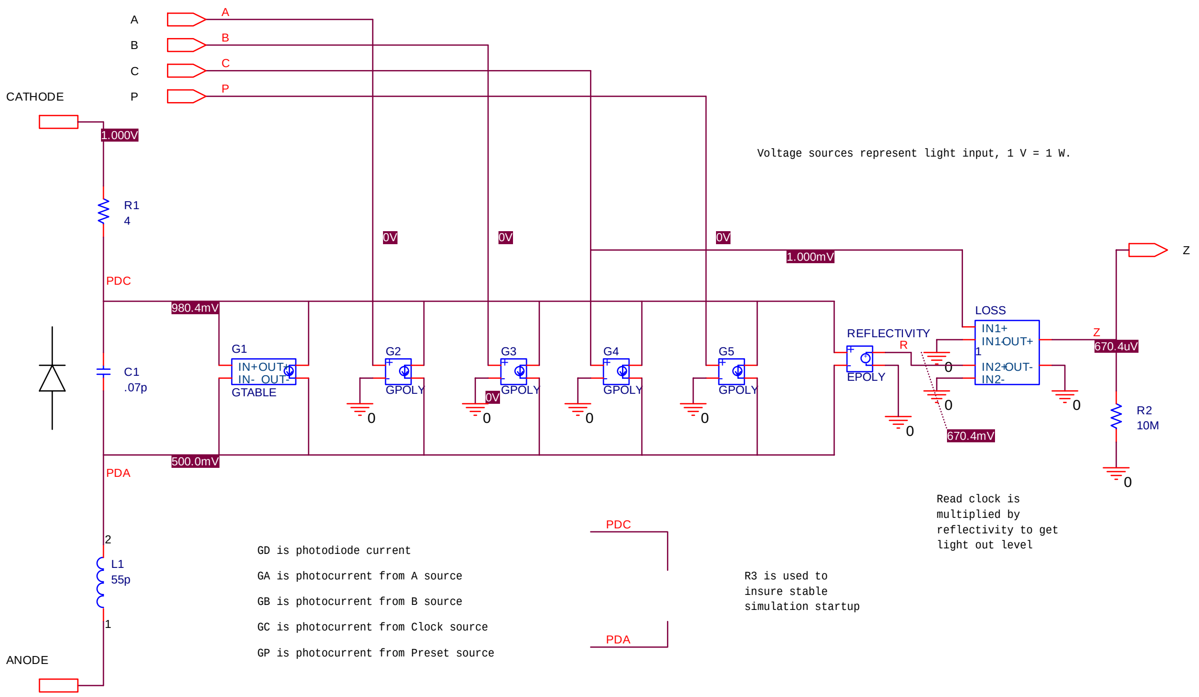
3.4. Reflectivity Curve

The reflectivity varies with the diode reverse bias voltage and can be represented quite accurately by a fifth order polynomial. This curve has been fitted to digitized data from an actual device in Figure 5. The data is given in Table 2. The equation is given by

$R = 5.72 + 2.06X + .28X^2 - .665X^3 + .042X^4 + .057X^5$.

3.5. Gate Operation

The data is read out using a clock beam, C. The output light, Z, is determined by multiplying the incoming light level by the reflectivity value, R.



Voltage sources represent light input, 1 V = 1 W.

Read clock is multiplied by reflectivity to get light out level

R3 is used to insure stable simulation startup

GD is photodiode current

GA is photocurrent from A source

GB is photocurrent from B source

GC is photocurrent from Clock source

GP is photocurrent from Preset source

Figure 2: Schematic of SEED Model

Table 1. Measured SEED current versus voltage for input to model.

Reverse Voltage (V)	Reverse Current (mA)
-0.8254	0
-0.8254	0.80851
-0.824	1.617
-0.80952	2.2979
-0.79365	3.3191
-0.78	3.7021
-0.75603	4.0851
-0.74016	4.4255
-0.71828	4.7234
-0.71428	4.9362
-0.66667	5.1489
-0.63492	5.234
-0.60317	5.2766
-0.53968	5.3191
-0.47619	5.3191
-0.44444	5.2766
-0.39682	5.234
-0.25397	5.1489
-0.031745	4.9362
0.19048	4.766
0.44445	4.5106
0.66667	4.3404
0.90476	4.1702
1.1429	4.0426

1.3651	3.9574
1.6825	3.9149
1.9841	3.8723

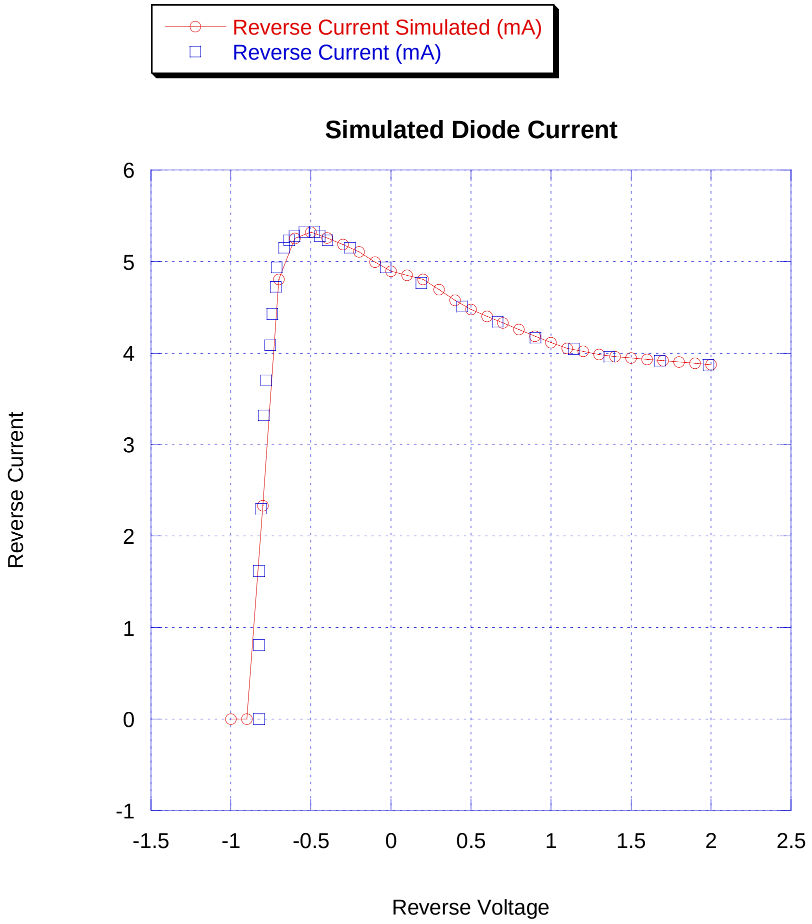


Figure 3: Simulated vs. measured p.i.n. diode current.

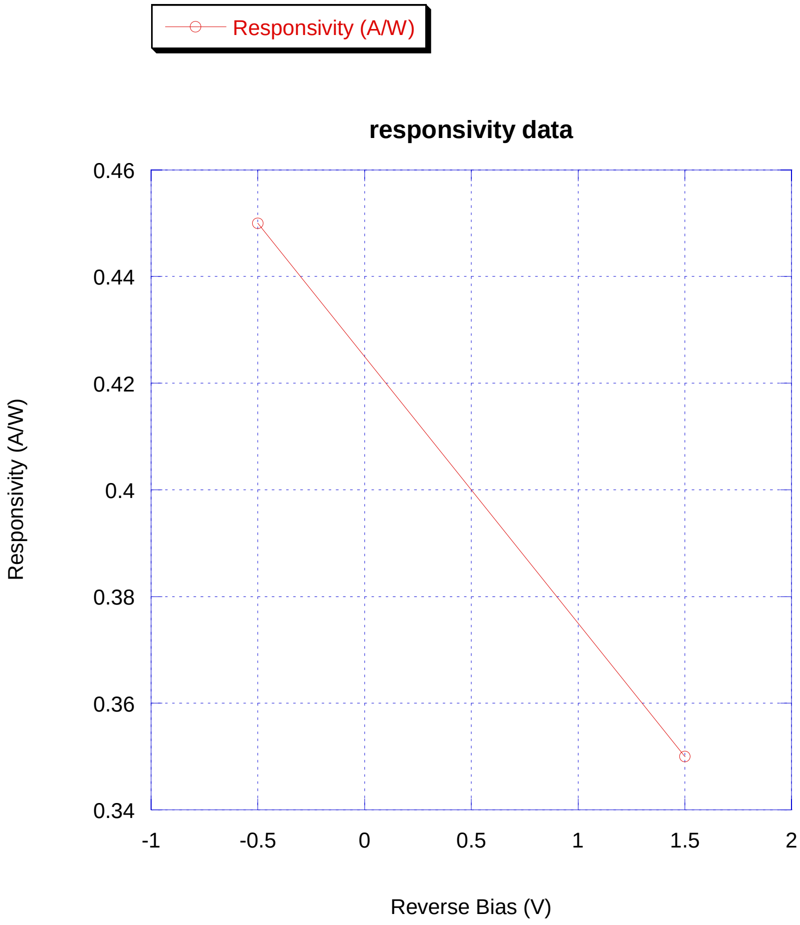


Figure 4: Responsivity curve.

Table 2. Measured reflectivity versus reverse voltage data.

Reverse Voltage (V)	Reflectivity (a.u.)
-1.00000	4.5714
-0.88889	4.5714
-0.82540	4.6032
-0.69841	4.6667
-0.60317	4.7302
-0.49206	4.8571
-0.39682	4.9841
-0.31746	5.1111
-0.25397	5.2381
-0.15873	5.3651
0.015874	5.7460
0.15873	6.0317
0.20635	6.1587

0.28572	6.3492
0.39683	6.5714
0.50794	6.7619
0.60318	6.9524
0.68254	7.0794
0.76191	7.2063
0.87302	7.3333
0.98413	7.4603
1.1111	7.5873
1.2222	7.6825
1.3492	7.7778
1.4762	7.8413
1.6667	7.9365
1.8889	8.0635
2.0000	8.1270

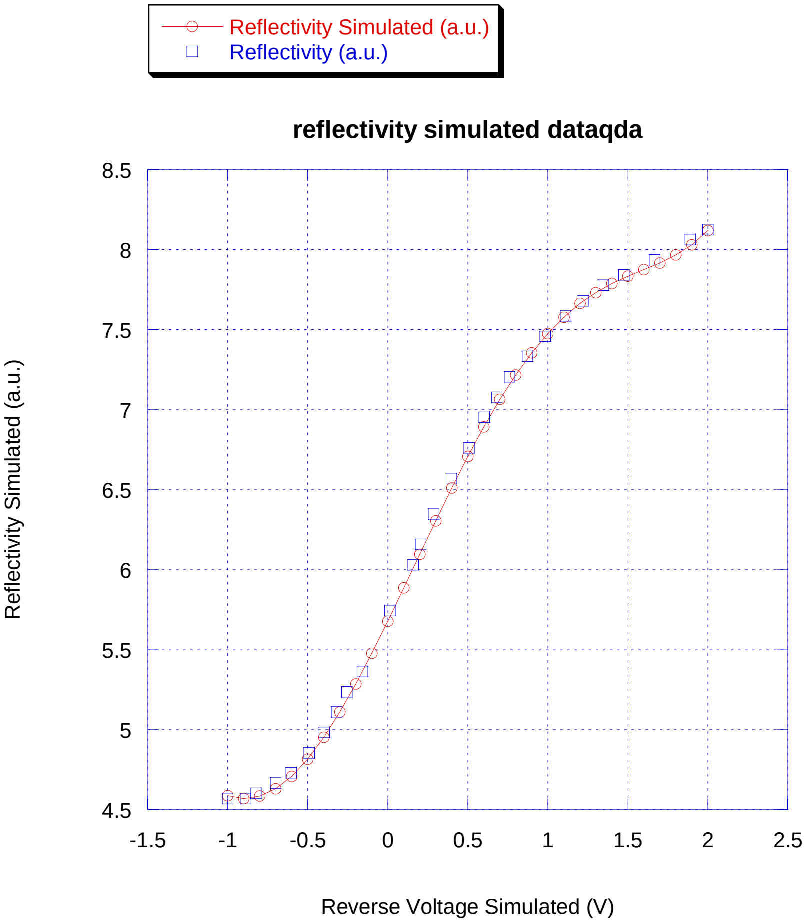


Figure 5: Reflectivity simulated data vs. measured.

4. S-SEED Switching Behavior

The switching characteristic of the S-SEED device was reported in a progress report [6]. The rise time of the S-SEED was 7ps with a settling time of over 75ps. The actual device is shown in Figure 6. The rise time was measured in Figure 7.

4.1. S-SEED Circuit

The circuit consists of two series SEED diodes (p.i.n. diodes are the two circular devices at the bottom of the picture) connected in parallel with a capacitor (the rectangular device at the top center of the picture) which was sized to provide transient current during switching. The two terminals were connected to a 1 V dc power supply. The PSpice schematic is shown in Figure 8.

4.2. Measurement Setup

The measured circuit has a train of light pulses, Pu1 and Pu2, which are shining on D1 and D2 respectively, at a 76 MHz rate. Pu1 and Pu2 are created from a common pump laser. Pu2 passes through a 1.8 meter delay line and arrives at the target delayed by 6 ns with respect to Pu1. Both pulses are vertically polarized by passing through a beam splitter. In operation, Pu1 discharges D1 and Pu2 discharges D2 6 ns later. In this way, the state of the logic gate is toggled continuously. The same beam splitter is used to create a horizontally polarized read signal, Pr1, which is itself delayed with respect to Pu1. The read signal is much smaller in intensity than either of the pulses so that it does not upset the state of the logic during the test. The read pulse is only applied to D2. The timing of these pulses is shown schematically in Figure 9.

4.3. Simulation Setup

The simulated circuit operates as follows. A 1 ps wide pulse, Pu2, is incident on D2 and discharges D2. The voltage across D2 approaches 0 V and, at the same time, the voltage on D1 approaches 1 V. A second 1 ps wide pulse, Pu1, is incident on D1 and discharges D1, thus switching the state of the circuit. The period between pulses is 200 ps. A constant read signal is applied to D2. The reflected output power of the circuit, POUT, is divided by the input power, Pr2, and scaled (by a factor of 10) resulting in a reflection coefficient measurement. The resultant waveforms are shown in Figure 10 and 11.

The output waveform has considerable ringing due to the series inductance in the device interconnect. By adjusting some of the model parameters, we can improve the device performance. For instance, if the series inductance is reduced from 55pH to 5pH, then the settling time improves from 100 ps to less than 10 ps. This also improves the rise time from over 5 ps to around 2 ps. There are two areas to target when reducing the series inductance in this experiment. First, the leads from the power supply probes can be shortened. Second, the interconnect between the two SEED devices can also be shortened. Building the circuit with an airbridge interconnect and widening the traces could lower the inductance.

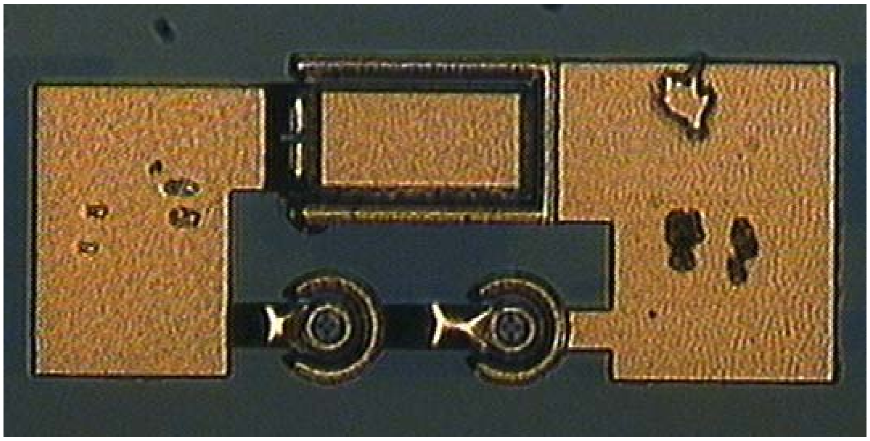


Figure 6: S-SEED photo.

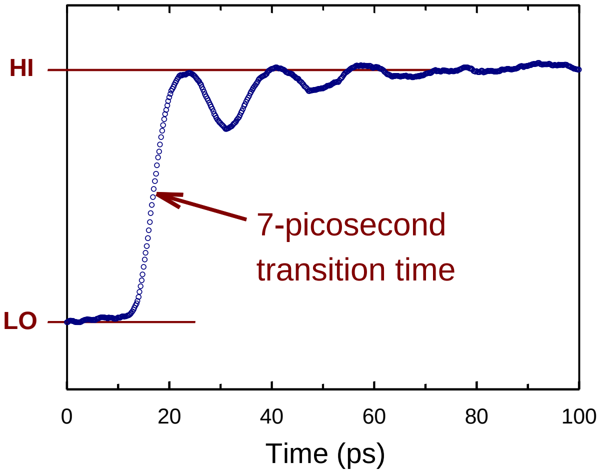


Figure 7: S-SEED switching characteristic.

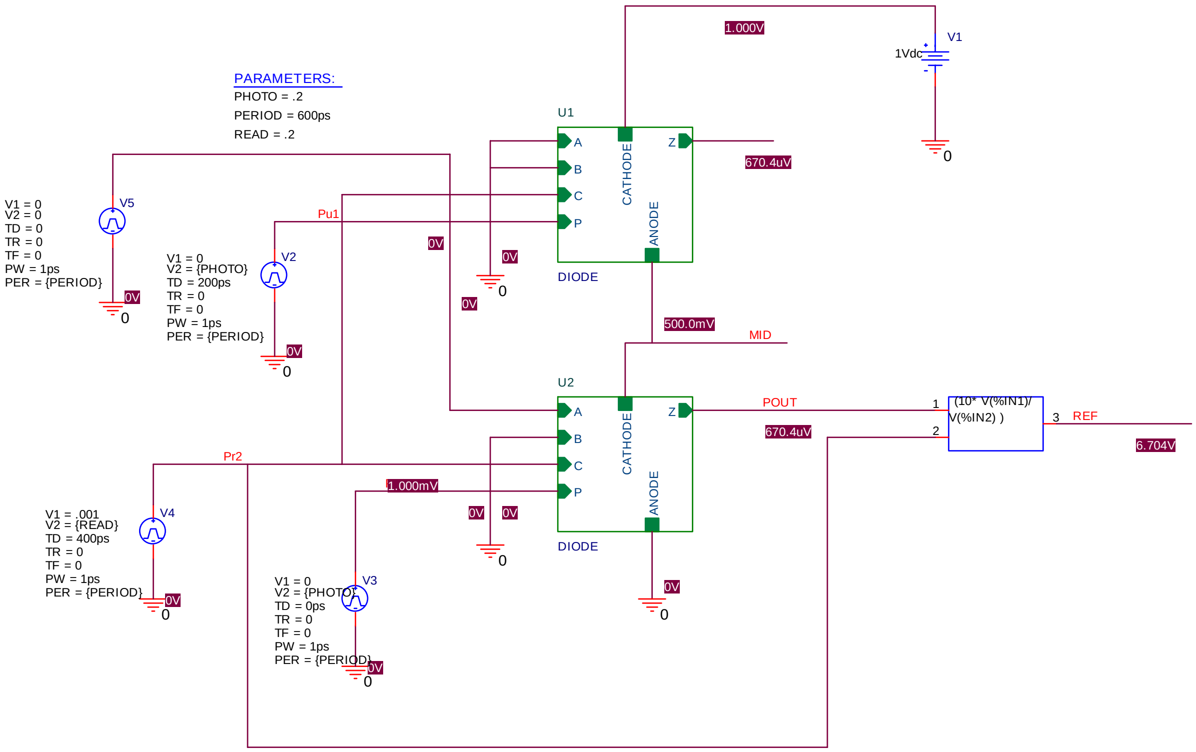


Figure 8: S-SEED switching speed PSpice circuit.

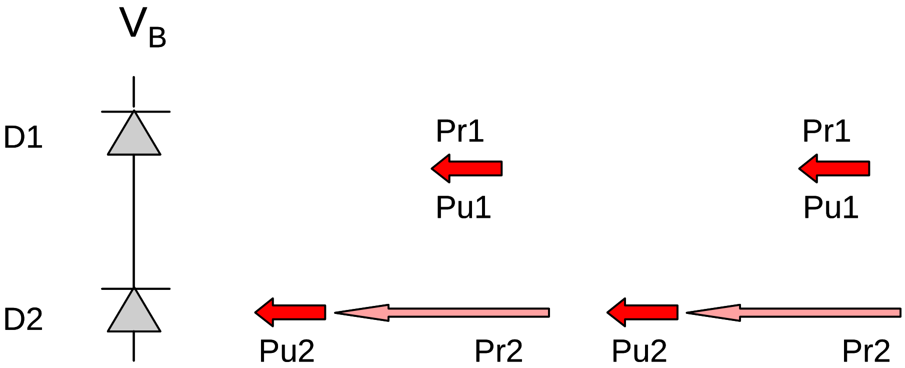


Figure 9: Pulse timing in S-SEED circuit.

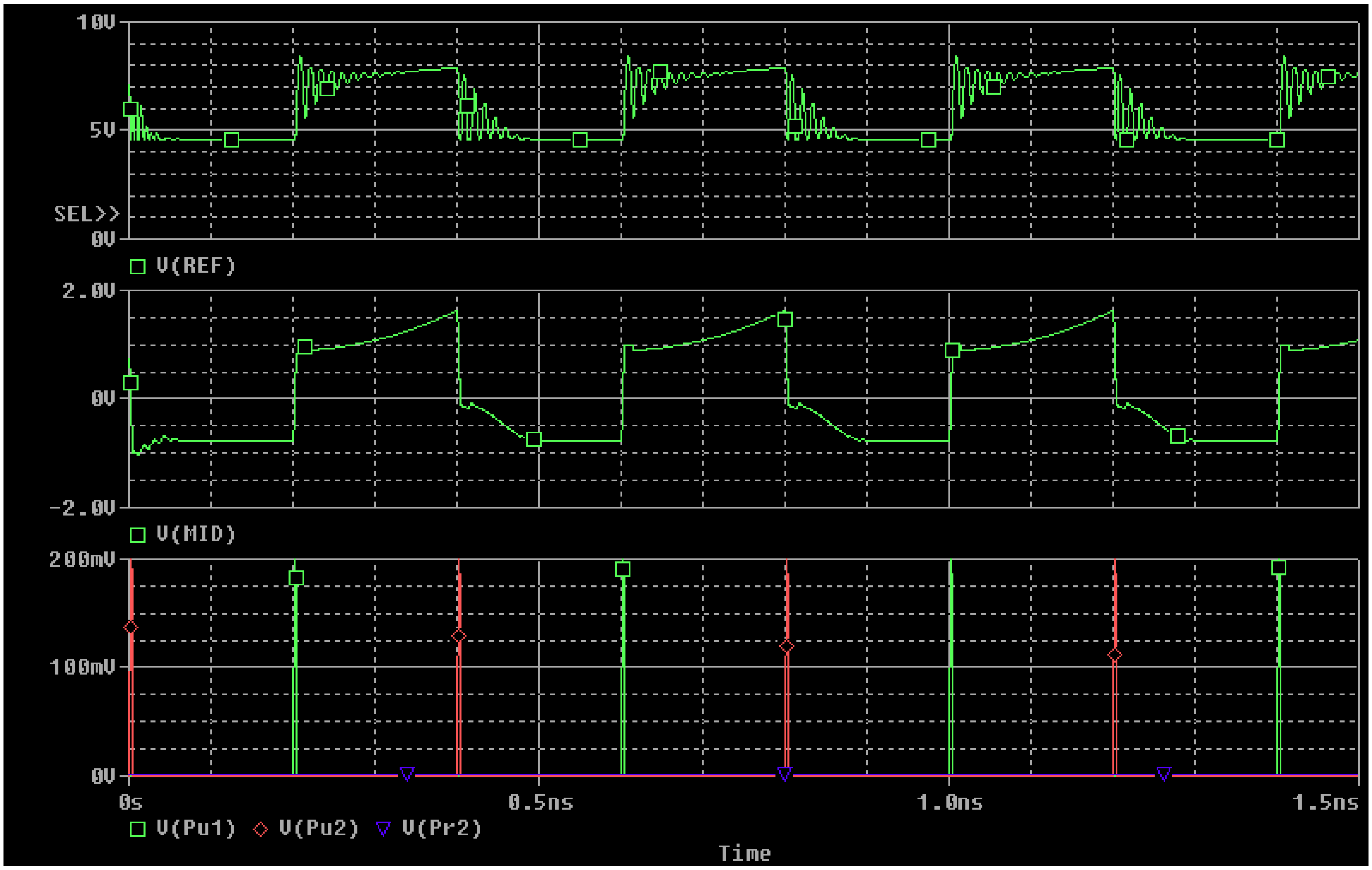


Figure 10: Switching speed simulation waveforms.

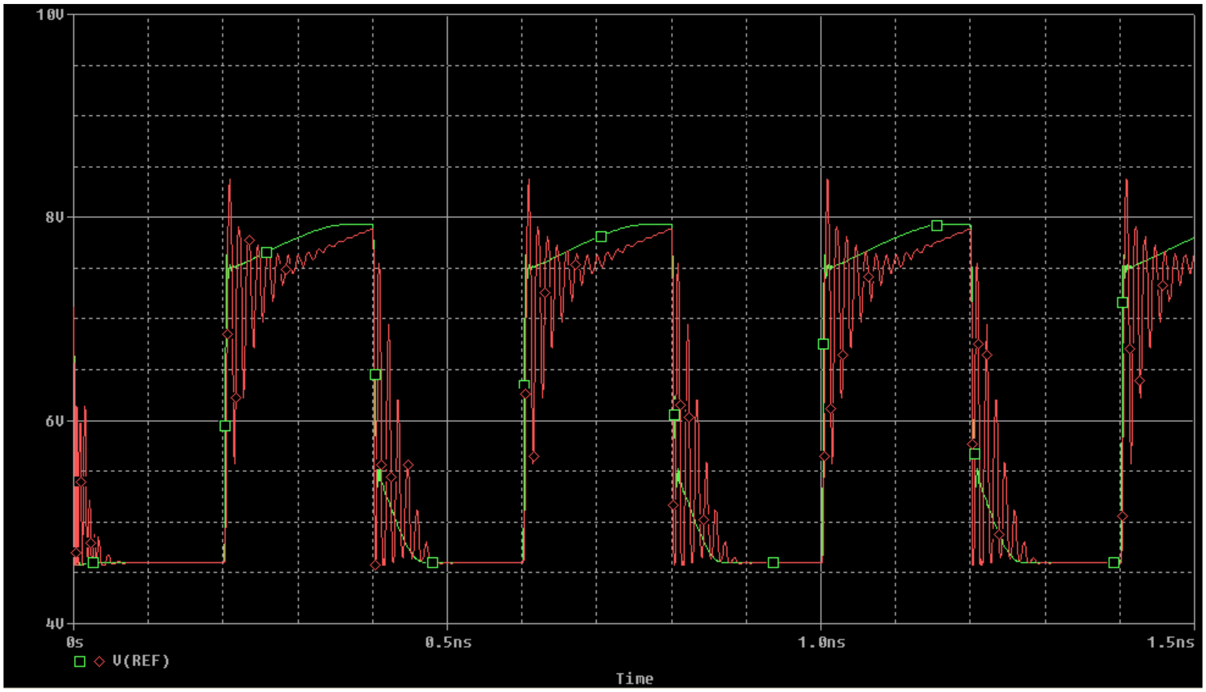


Figure 11: Effect of reducing series inductance.

5. Characterization of the Optical Gate

Each digital logic gate will have a transfer characteristic and switching characteristic that describes the output waveform in terms of the input. In the case of the optical SEED logic gate it is more difficult to measure because the input and output signals are pulses of light. A series of simulations were used to measure the switching and transfer characteristics.

The simulation consists of an OR gate configuration (similar to Figure 18) with a single data input. The schematic is shown in Figure 12. The Preset input is used to discharge D1 (charging D2) and raising the voltage on MID. The data input, D, is pulsed to a maximum value which switches the gate and the magnitude of the output pulse is measured at POUT and POUTB during the clock pulse. The DB input (invert of D) has a maximum value that is .5625 * Dmax. This value represents the extinction ratio of the optical gate. The reflection of the clock input, C, is a function of the diode bias voltage ranging from a minimum of .45*C to .8*C. Since D is the true logic value, DB should be relatively dimmer by the extinction ratio.

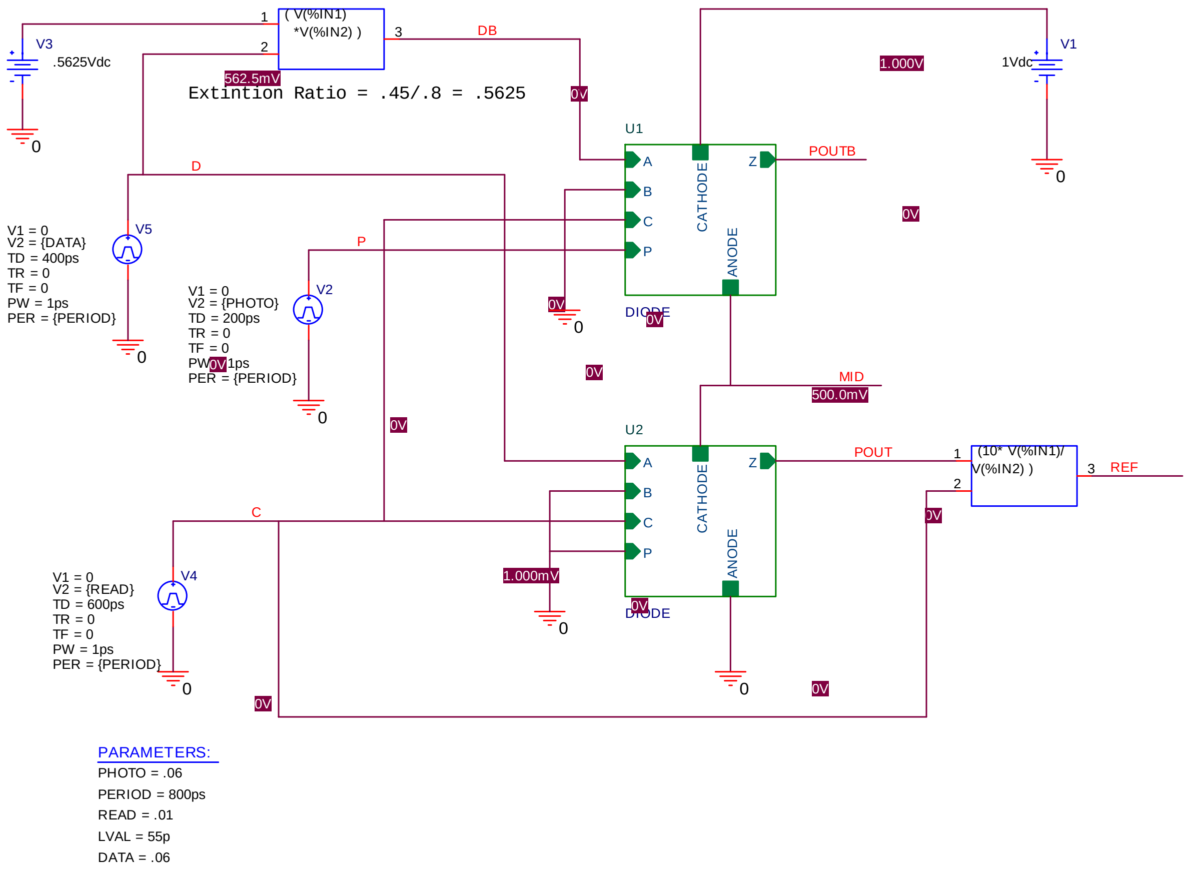


Figure 12: SEED Logic gate transfer characteristic schematic.

The circuit in Figure 12 was used to characterize the switching of the optical logic gate. The P input was used to alternatively switch logic states. The voltage on the electrical output MID varied from a high of 1.75V to a low of -0.75V. Depending on the amplitude of the input pulse, the signal exhibited overshoot. The input pulse was varied from 20uW to 70 uW and the rise/fall time of the pulse was measured. These values are given in Table 3. The goal was to find a pulse height that would generate just enough current to charge the alternate capacitor (of the diode) without having too much current and overcharging the capacitor. The result was that the 60 uW pulse appeared to have the fastest switching time without excessive overshoot exhibited with larger pulses. All pulses were 1ps in width. The switching characteristic of the MID node for 60 uW pulses is shown in Figure 13. This was the chosen pulse magnitude for future circuit design.

Table 3. Optical gate switching speed.

Pulse Height (uW)	Rise Time (pS)	Fall Time (pS)
20	no switch	no switch
30	no switch	no switch
40	100	100
50	33	32
60	23	23
70	20	20

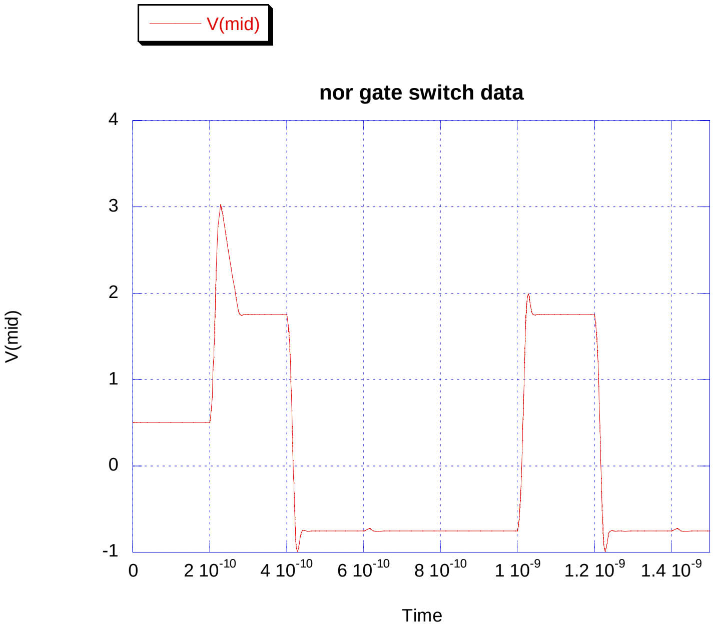


Figure 13: NOR gate switching with 60 uW pulses.

An actual logic input switches the gate differently than a preset pulse because the pulses are generated by previous logic gate for which the inverted and non-inverted outputs do not turn off completely. The low output is only 45% of the clock input while the high output is 80% of the clock input. An extinction ratio of about 56% is the result. Therefore, another series of simulations were performed to characterize the switching of the gates logic state by real optical inputs.

For a 60 uW pulse height on data and clocking inputs, we simulated the amount of preset needed to effectively “set” the state of the SEED device. We swept the preset pulse energy from 10 uW to 90 uW. Again, the switching characteristic for the MID node was examined with the goal of having enough current to charge the capacitor yet not overcharge it. The resulting data in Figure 14 shows the least amount of overshoot on v(mid) that effectively "presets" the state of the device was 32uW.

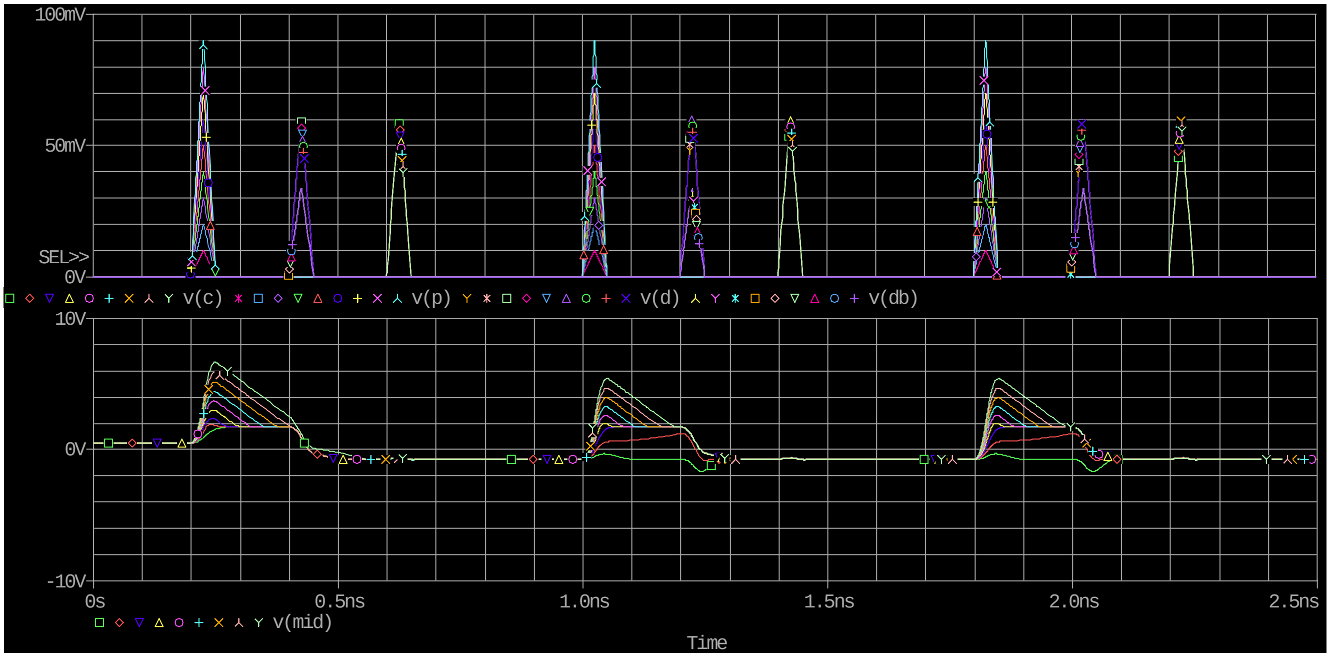


Figure 14: Switch point simulations

A preset height of 40 uW was used in future designs. Now that we determined how much preset was required to set the state of the device, we varied how much clock power was needed to read the state and effectively set the state of the next cascaded device. We used 40 uW of preset and swept input power from 50 uW to 90 uW. The constraint was that the amount of output power had to be at least 32 uW. The result in Figure 15 showed that at least 75uW of clock power is needed to adequately cascade into the next stage and "set" the next gate. Future design and simulations will use minimum preset, data, clock powers of 40 uW, 32 uW, and 80 uW.

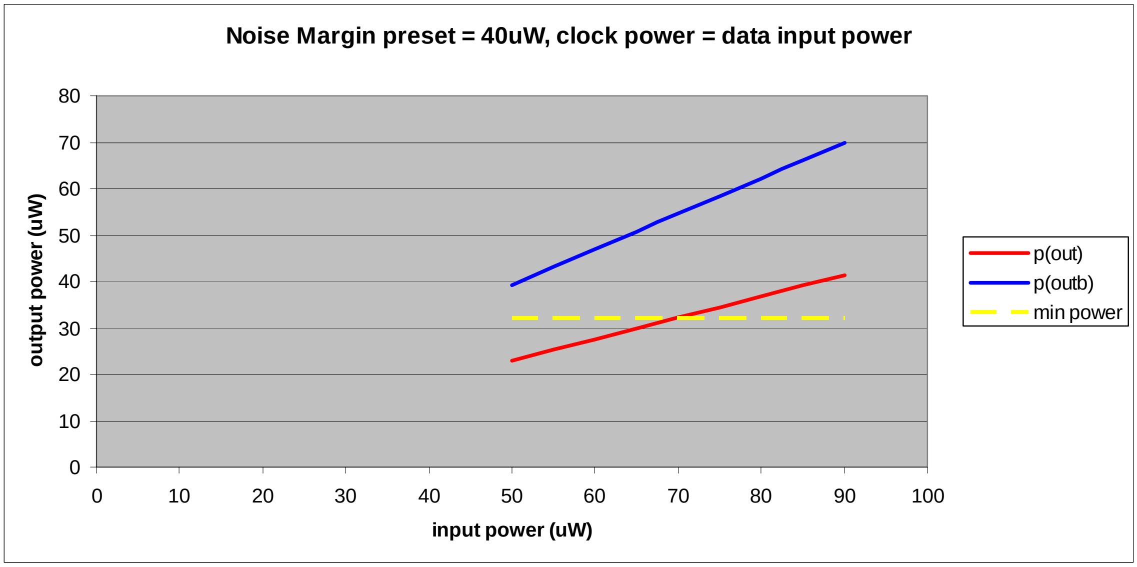


Figure 15: Simulation rules for input/output power.

We demonstrated a simulation of gates cascaded three deep in order to show outputs of one gate effectively setting the state of the next while clocking out with enough energy and minimal overshoot. Figure 16 shows the design for cascading of logic gates. We examined V(MID) on each of the three gates for overshooting and whether or not it switched. Figure 17 shows that each of the stages is effectively switched with minimal overshoot.

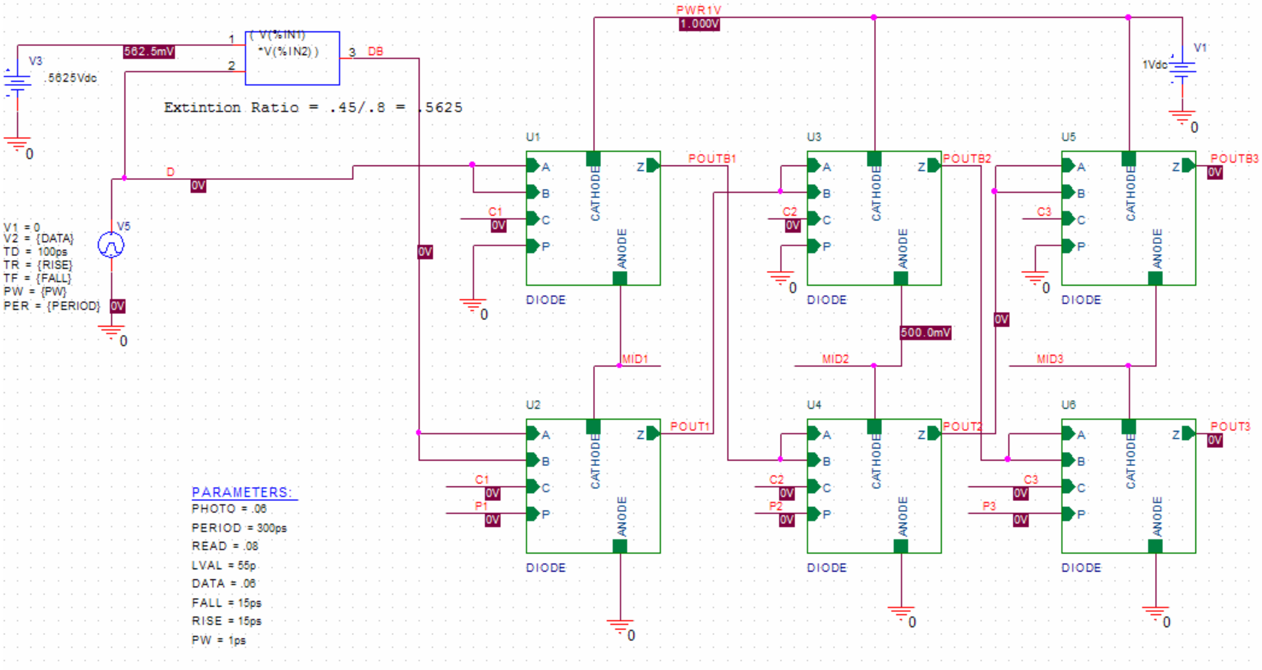


Figure 16: Cascading of logic gates 3 deep.

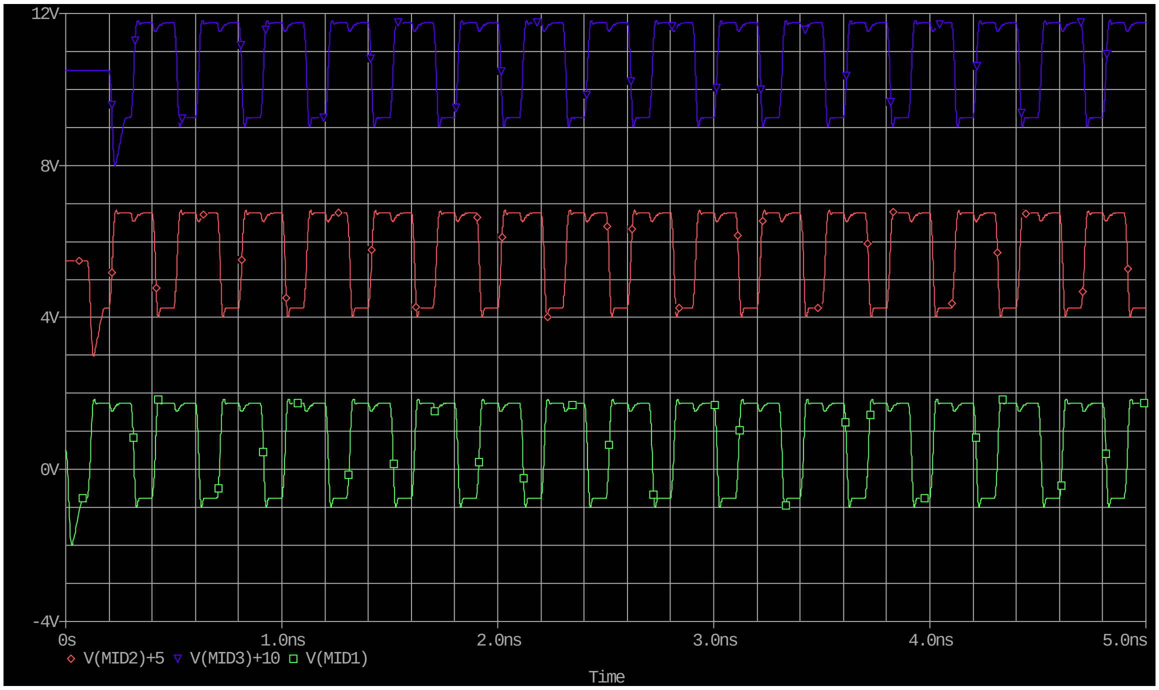


Figure 17: V(MID) for each SEED pair.

6. Optical Logic Gate Simulation

A simple two input optical gate was designed and modeled from the S-SEED device models. The schematic of an OR gate and an AND gate are shown in Figure 18 and Figure 19 respectively. The two diodes are connected in series with a 1 V power supply in reverse bias. Inputs A, B, AB, BB, C and P and outputs Q and QB are optical inputs and outputs simulated by PSpice as voltages where 1 V = 1 uW optical power. Inputs AB and BB are inverted versions of the inputs A and B. MID is the electrical node connecting diodes D1 and D2.

The optical logic gate operation is based on a three phase clocking scheme shown in Figure 20. The following describes the operation of the OR gate. In the first phase, the logic of the gate (OR) is set using the P input. The P input will discharge D1 and charge D2. The voltage on node *MID* will rise toward VCC. In the second phase, the logic inputs (A, B, AB, BB) are applied to the two diodes. Unless both AB and BB are ON, the gate will not switch from the initial state set in phase 1. This performs the logic function. In phase three, the C input is applied to both diodes, and the light reflecting from the Q (charged diode) indicates the resulting of the logic function. The clock input is equal on both diodes and will not switch the state of the gate by itself. The simulated output of this logic gate is shown in Figure 21.

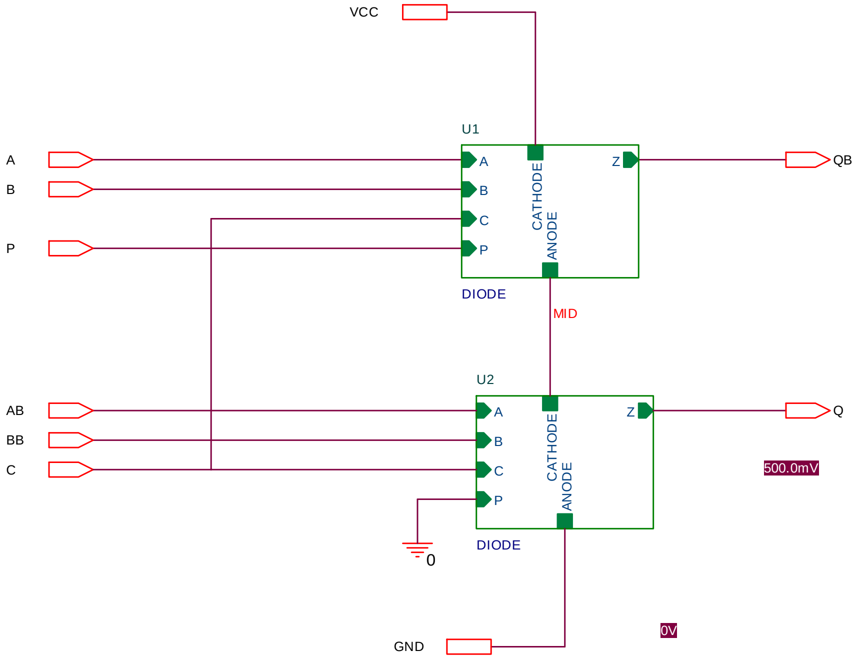


Figure 18: Two input optical OR gate.

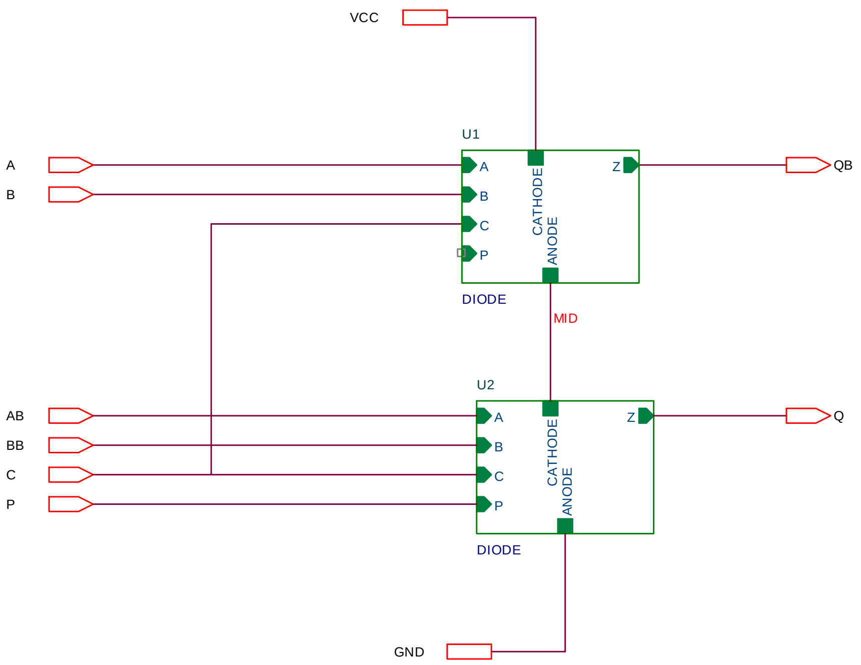


Figure 19: Two input optical AND gate.

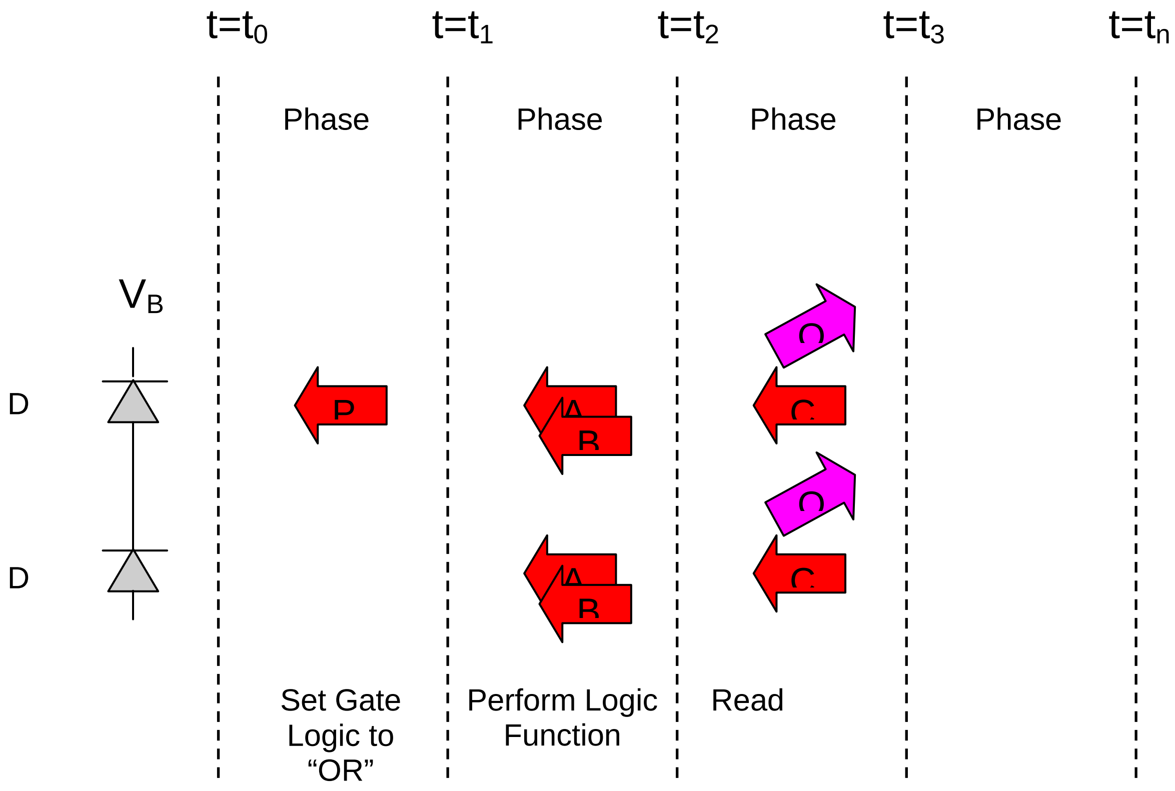


Figure 20: Three phase logic diagram.

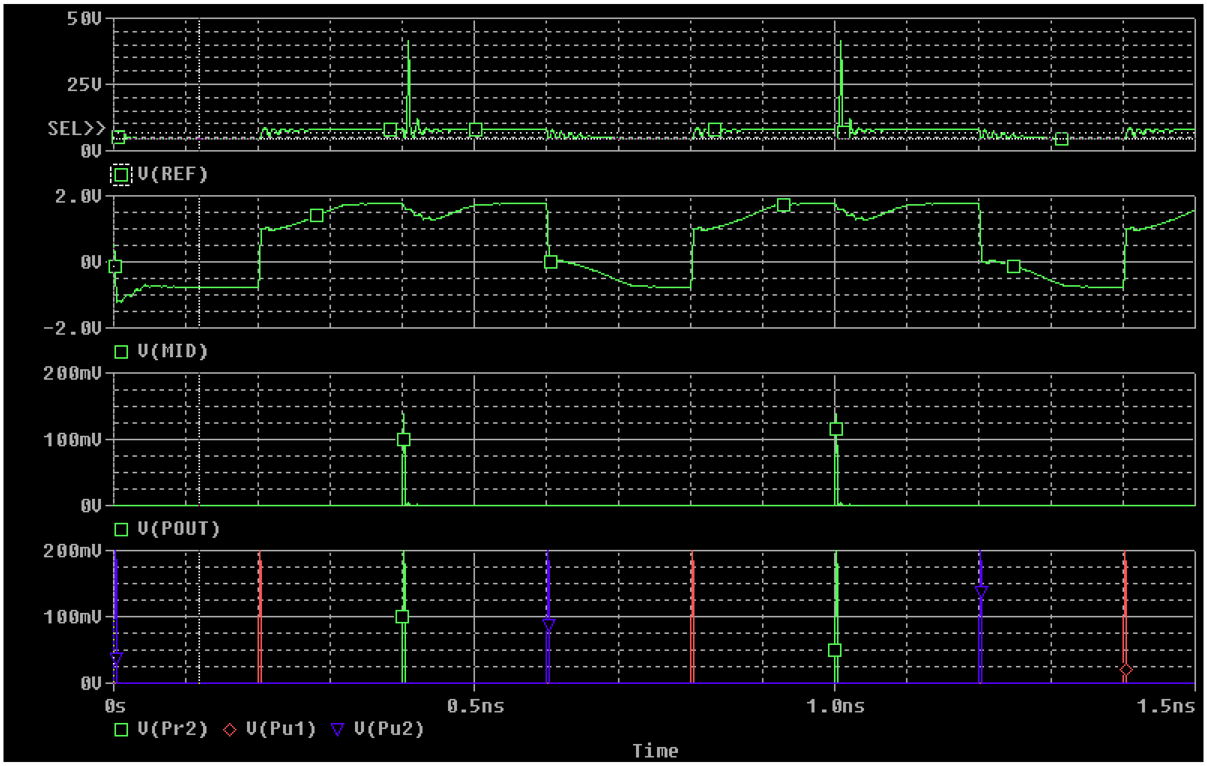


Figure 21: Simulated photonic logic gate operation.

In order to demonstrate AND, OR, and XOR gate functionality, we built a two bit counter that delivers differential logic HI and logic LO. Figure 22 shows the functional block takes input from a piece-wise linear voltage source that describes pulse widths and heights. These inputs are then multiplied by a two bit pattern to generate 00, 01, 10, or 11 in appropriate time scale.

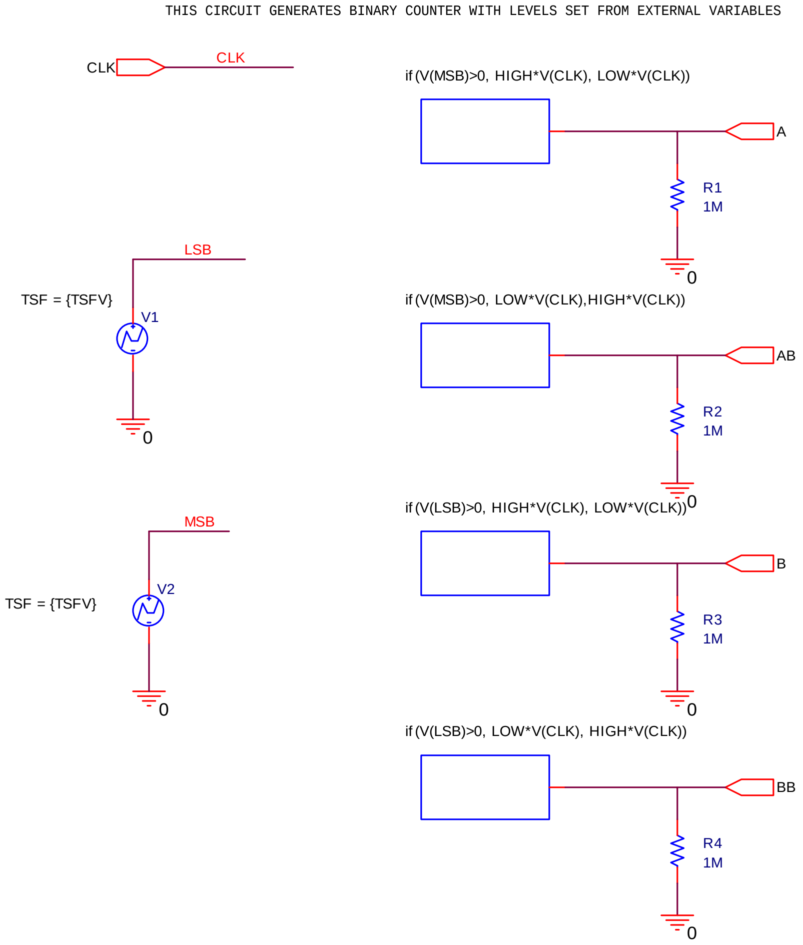

THIS CIRCUIT GENERATES BINARY COUNTER WITH LEVELS SET FROM EXTERNAL VARIABLES
CLK
CLK
if (V(MSB)>0, HIGH*V(CLK), LOW*V(CLK))
A
R1
1M
0
LSB
TSF = {TSFV}
V1
0
if (V(MSB)>0, LOW*V(CLK),HIGH*V(CLK))
AB
R2
1M
0
MSB
TSF = {TSFV}
V2
0
if (V(LSB)>0, HIGH*V(CLK), LOW*V(CLK))
B
R3
1M
0
if (V(LSB)>0, LOW*V(CLK), HIGH*V(CLK))
BB
R4
1M
0

Figure 22: Two bit counter functional block

With this two-bit counter functional block, we demonstrated functional operation on AND, OR, and XOR gates in the figures 23, 24, 25, 26, 27, and 28.

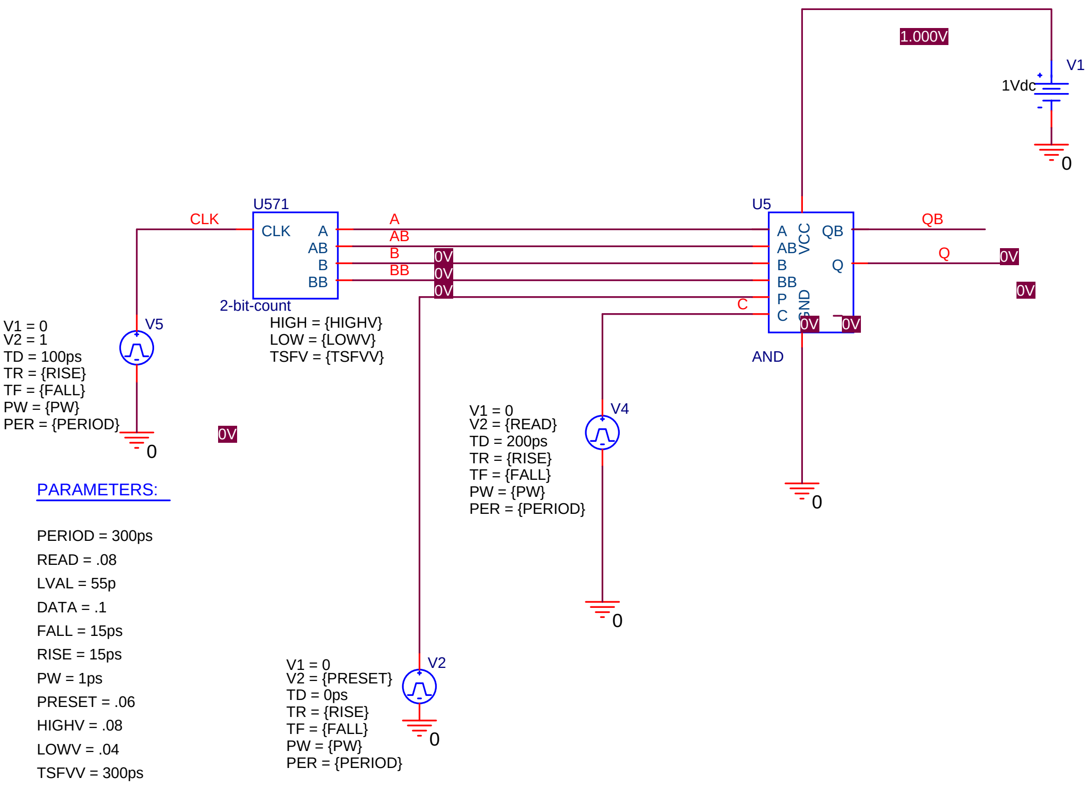


Figure 23: AND demonstration circuit

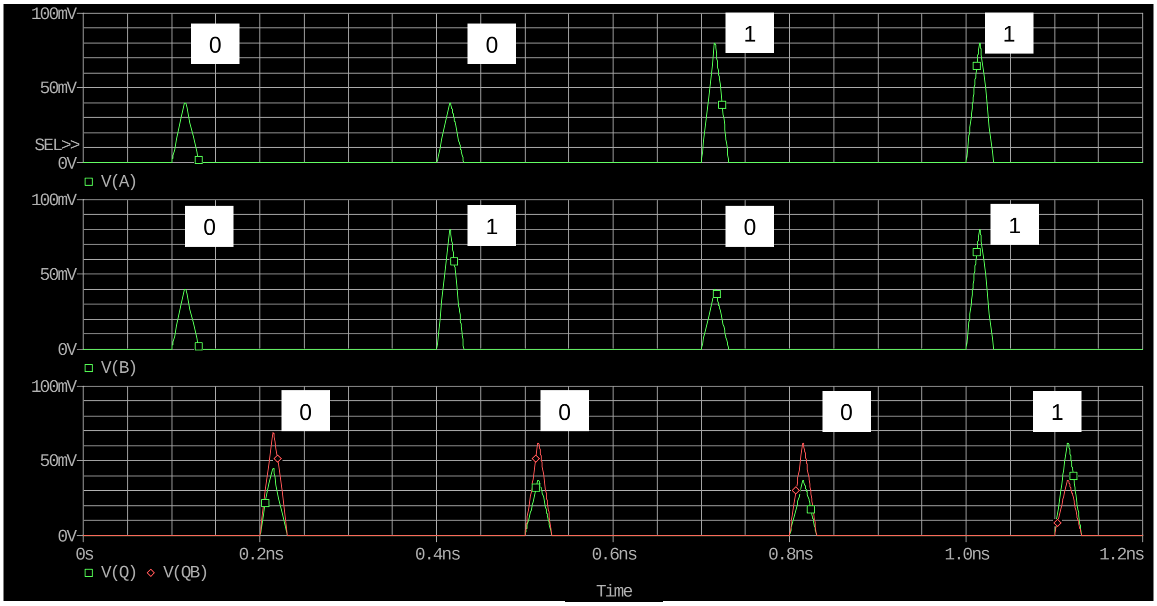


Figure 24: AND demonstration results

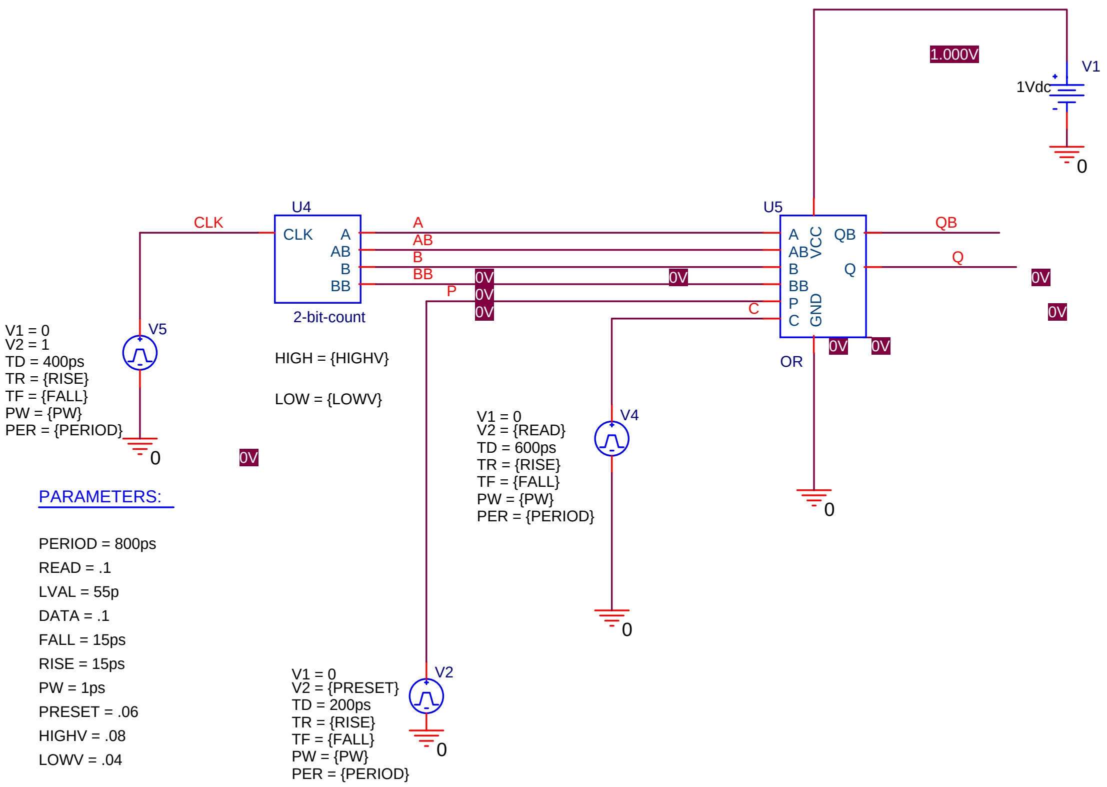


Figure 25: OR demonstration circuit

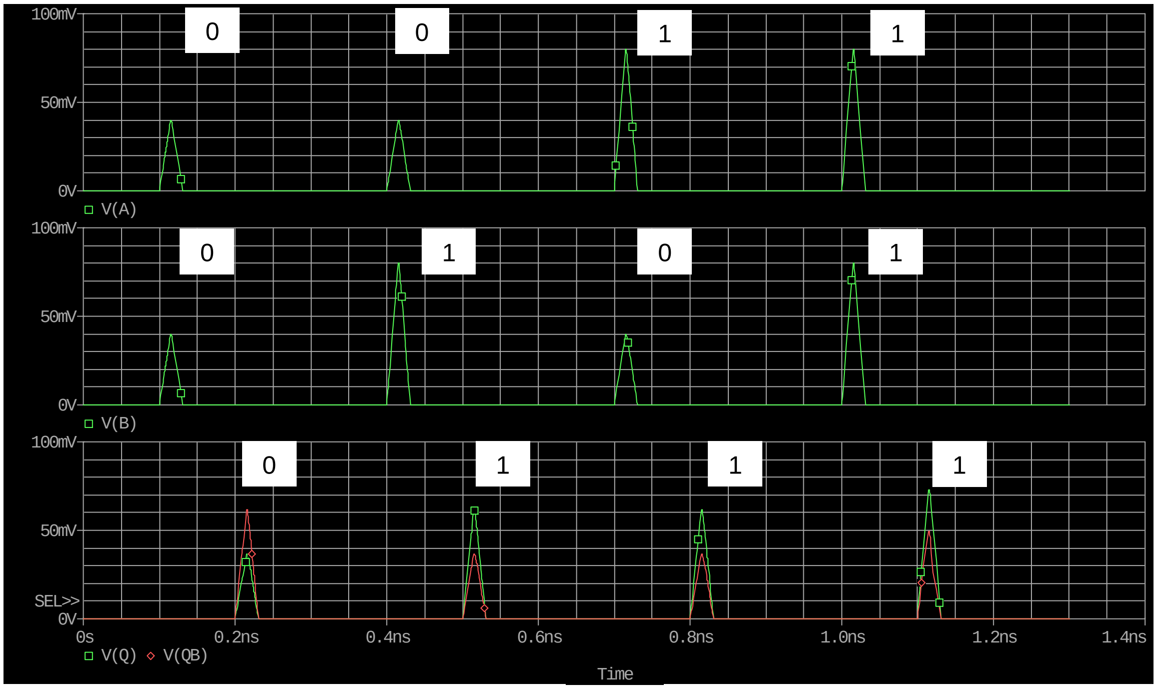


Figure 26 OR demonstration results

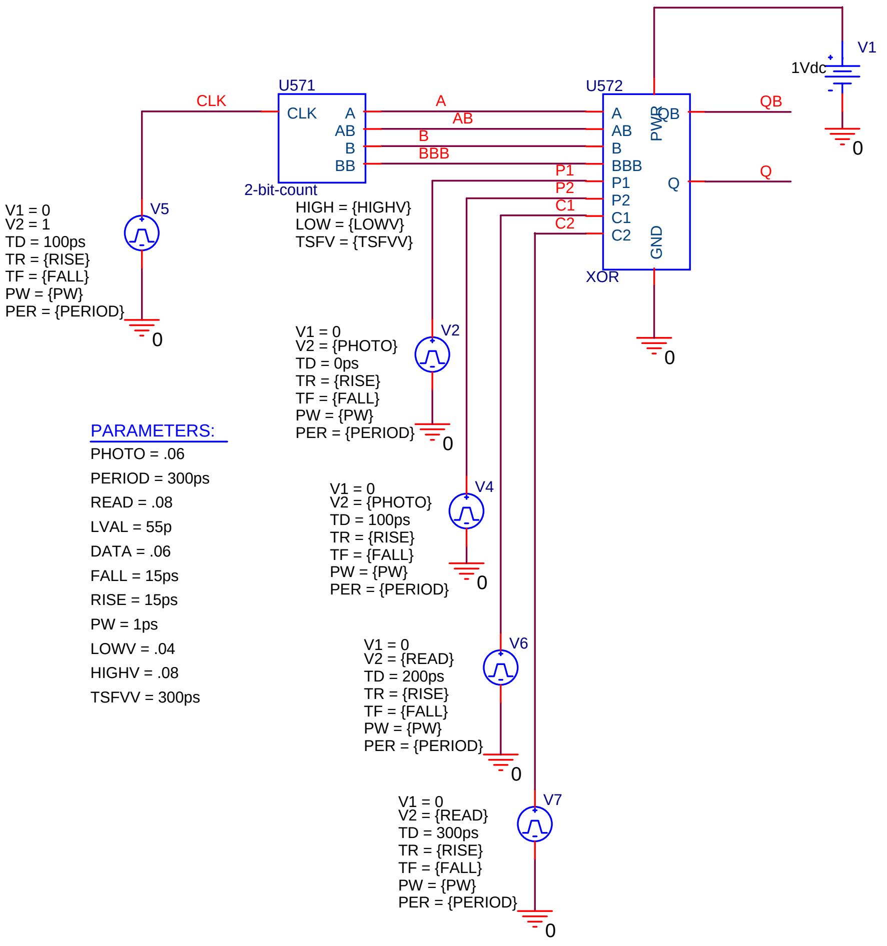


Figure 27: XOR demonstration circuit

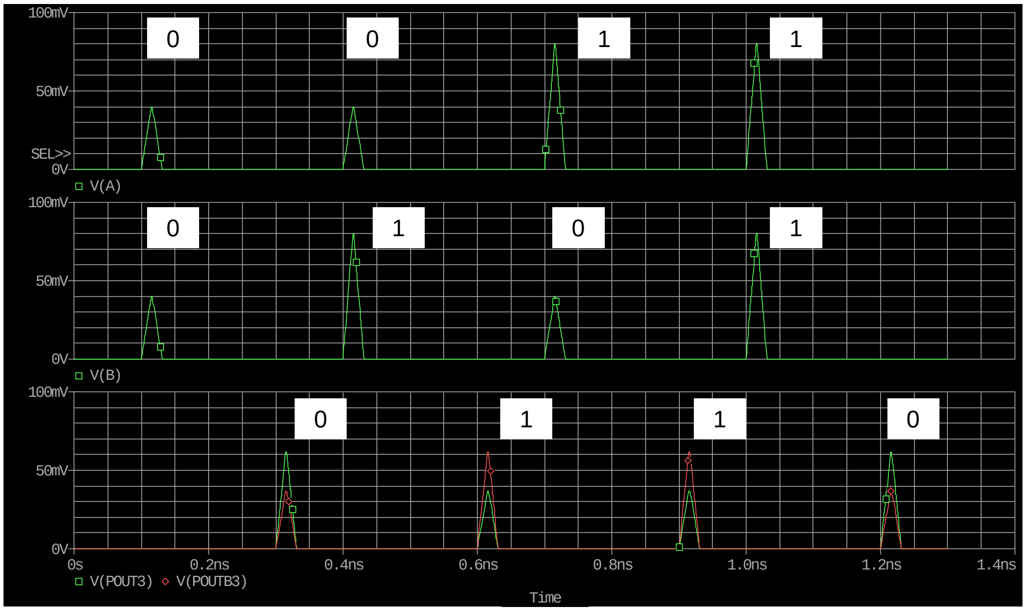


Figure 28: XOR demonstration results

7. Stream Cipher for Optical Communications Systems

7.1. Why a new stream cipher?

We examined existing ciphers, but found nothing that looked buildable within our design constraints. Existing block ciphers like AES and DES require thousands of gates to implement. The best stream ciphers contain long, stuttering, shift-registers, clocked in an irregular pattern. This style of logic seems very doubtful when contrasted with our current capabilities. While we eventually hope to implement arbitrary computation capabilities, this will be a ways down the development road. The new proposal has the important advantage of buildable. Moreover, it can be built up a little at a time, and adding additional stages should be easy.

7.1.1. The Optical Cipher Concept

The new design sends the plaintext message as a bit stream through a series of very simple ciphering boxes. The security of the system comes from using a large number of boxes. A smaller number of boxes can be used as a data scrambler.

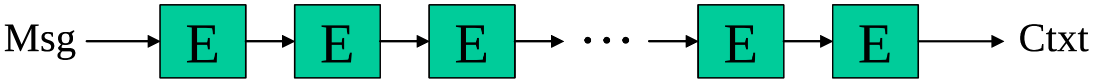


Figure 29: Optical Cipher

In the individual ciphering stages shown in Figure 29, the E-boxes perform very simple operations on the data stream. A simple logic function is computed using a few bits from the data stream, and the single-bit result is Xored into the upstream data.

The inside of an E-box is shown in Figure 30:

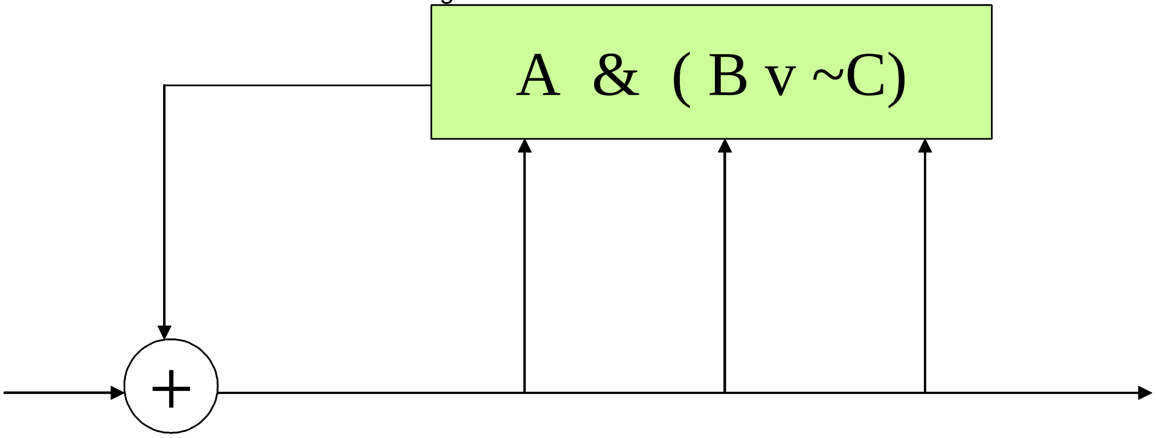


Figure 30: E-box internals.

In Figure 30, the logic function is

F(A,B,C) = A or (B and not C).

We will probably use a few different kinds of logic functions. In order for this to be decodable, the bits used as input to the logic function must be older than the bit to be modified, so the taps are taken from the ciphertext side.

Decryption is very similar, but not identical. There is a series of D-boxes, applied to the ciphertext. Each D-box similar to Figure 31 undoes the effect of one E-box. The D-boxes are used in the reverse order of the E-boxes.

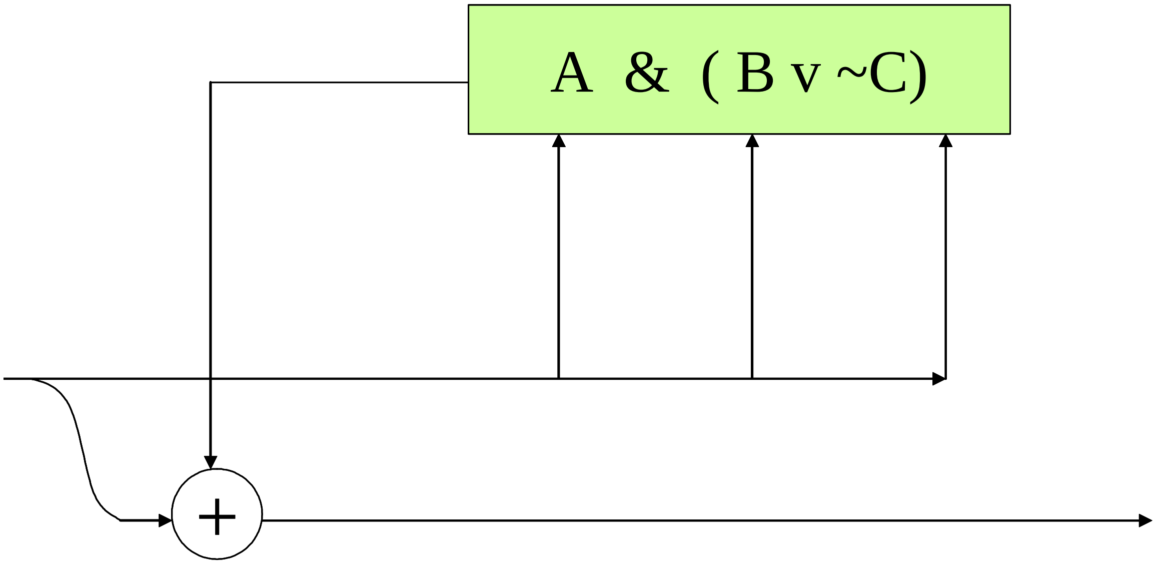


Figure 31: D-box internals.

7.1.2. Circuit Variations

There are several variations possible on the basic theme. We could use D-boxes in the Encryption chain (and match them with E-boxes in the Decryption chain). This loses the Finite-Error-Propagation property. We could also use a more complex arrangement of E-boxes, with some E-boxes wrapped around others, or with interleaved inputs. As long as the inputs to E-boxes only affect newer data stream bits, the resulting output can be decoded. However, the decoding becomes more complicated. This option complicates the device construction with no obvious improvement, so we didn't pursue it.

7.2. Keying

Keying is done by making small changes to the logic function. The inner logic function is fixed, but the inputs and outputs can be modified in simple ways. The E-boxes in the transmitter and the D-boxes in the receiver must use the same keying information.

There are lots of possible keying options as seen in Figure 32.

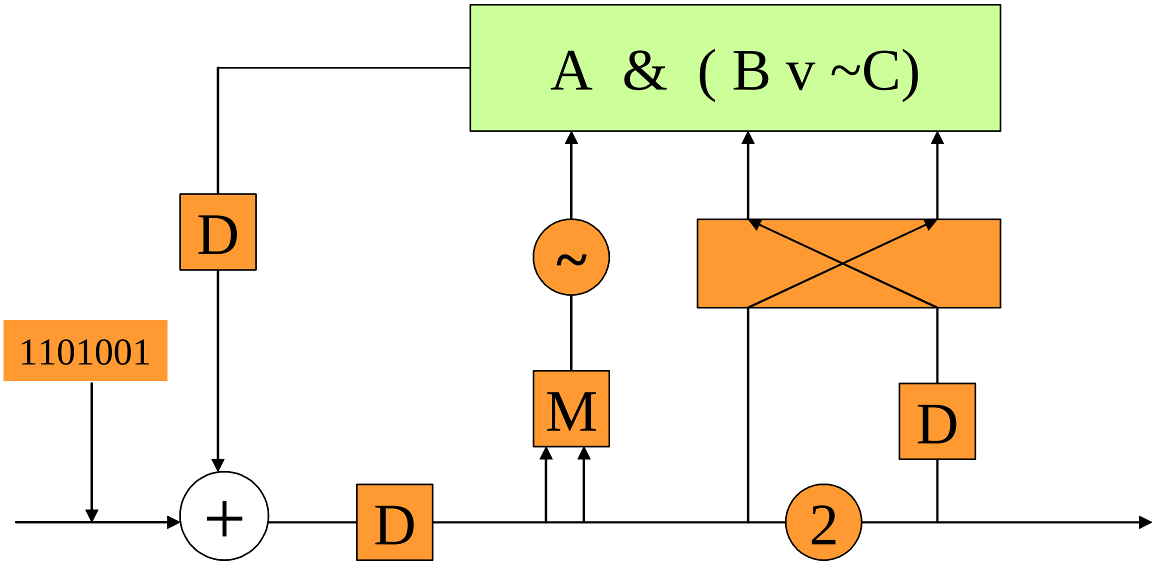


Figure 32: E-box with keying options.

In Figure 32:
D represents a delay of a few bit times;
~ is a Not gate;
Mux selects one of two inputs;
X exchanges two inputs of the logic function;
2 exchanges two bits in the data stream;
1101001 represents a random initialization vector.
In the final system, only a couple of these options will be used.

Keying is accomplished by turning these various kinds of boxes On or Off. In the Off state, they simply pass data through; in the On state, they do their particular modification. There are a few other keying ideas we considered and dropped.
These include:

(a) using key to select from among two or more F functions; this was discarded because it means there's always unused logic;
(b) using key to vary the data path of the encryption stream, so the E-boxes can be used in various orders. This is simply hard to build.

7.3. Keying Notes

There is an external user-supplied key. This key should be at least 64 bits for a respectable cipher, but could be smaller for a data scrambler. The key is processed by a control computer, using a hash function like SHA-1, to create (more) internal keying bits. For real encryption, there must be at least 128 internal keying bits. These bits are supplied to control the various keying options of the E-box inputs. The key can be supplied, and changed, at electrical speeds. A given key will be used for at least a second; during key switchover, a period of nano or microseconds, the output of the encryption device will be suppressed. Each module can use only about 4 or 5 keying bits, so we'll need a minimum of 25 E-boxes for encryption. More is probably better, so we might use as many as 50 E-boxes.

7.4. Finite Error Propagation

The design has the property that a transmission error will cause only a finite amount of damage to the downstream decryption. After the error has passed through the decryptor, the effects will circulate in the internal state for a while and then be damped out.

7.5. Simulator

We've written a simulator for the Encryptor. The simulator can encrypt or decrypt a data stream. The current version of the simulator can use up to 1000 E-boxes, in a fairly complicated pattern. It contains several methods of measuring the apparent randomness of a data stream. The simulator has been used in a number of experiments to try out various design details.

7.5.1. Logic Function Effectiveness

The position of the F-function within an E-box can be taken by various different logic functions. In the pictures above, we've used the function F(A,B,C) = A v (B & ~C), but many other functions are possible. We are interested in evaluating the cryptographic effectiveness of different logic functions, and comparing this to the construction difficulty ("cost") of the functions. Cryptographic strength is hard to measure. We use a simple proxy measurement, suitable at this stage of our design: Statistical randomness. (Later on, as we fix more details of the design, we'll do the more difficult measurements of crypto strength.) Statistical randomness is evaluated by taking a simple input signal such as a stream of all 0s, and looking at the distribution of various patterns in the output stream.

In a typical experiment, a stream of 1,024,000 consecutive 0 bits is supplied as the input.
The output is sampled with a sliding 8-bit window, and a histogram of the 256 possible output values is created. A random output will have about 4000 instances of each possible value. The difference from random is evaluated by looking at the standard deviation of the values. A sample run is shown below. This run uses 75 modules.

```
[r@fermat r]$ optenc2a 75 x x x
optical encryptor simulation
  3719  4149  4106  3997  3945  3992  4070  3962
  3957  4106  3937  4020  4201  3836  4147  3907
  3980  4035  4193  4114  3964  4113  3830  4189
  4030  3947  3791  3948  4083  4107  3808  4020
  4029  4020  4151  3999  4241  4154  4325  3944
  3905  4152  3930  3915  4048  4031  3882  4205
  4012  3951  3756  4094  3843  3968  4087  3890
  4024  4020  4050  4031  3847  3994  3966  3934
  4027  3991  4009  3916  4059  4040  3822  3817
  4242  4176  4053  4253  4110  4125  4024  3811
  4081  3839  4135  3890  3904  4071  3947  3886
  4006  4033  4175  3936  4030  3801  4377  3939
  3913  4086  4176  3942  3991  3646  3937  4083
  3765  4187  3933  4209  3999  4198  3844  3940
  4119  4085  4036  3771  3866  3935  4033  3818
  4123  3829  3951  4196  4050  3929  3881  4076
  4149  3954  3831  4035  4118  3965  3967  4092
  4058  4201  4140  3999  3776  3903  4043  3921
  4069  4115  4202  4155  4093  3732  4249  3898
  3933  3903  4020  4029  3961  3974  4033  3880
  3989  3905  3948  3640  4177  4152  3910  3891
  4015  3873  4045  3918  3991  4080  3949  4111
  3987  4167  3881  3926  4109  4174  4110  3894
  4180  3787  3751  3820  4105  4153  4013  4023
  4076  3875  4074  4143  4200  4099  3857  4147
  3942  4181  3772  3894  3726  3924  3911  4102
  3813  3749  4194  3911  3984  3892  4124  4174
  4148  3774  4108  4068  3937  3770  3881  4097
  4038  4131  4123  4062  4132  4020  3713  3930
  3797   3918   3943   4089   3923   3978   3863   4038
  4050   4100   4116   3872   3849   4097   3868   4083
  4027   4159   3995   3755   4136   3821   4076   4006
max = 4377  min = 3640  max-min = 737
sum = 1024000  avg = 4000.000000
sumsq = 4100429126  sigma2 = 17301.273438  sqrt = 131.534305
75 modules
```

The standard deviation in this example is 131.5. In a uniform random distribution, it will be roughly the square root of the bin average. For these runs, the bin average is 4000, and the SD should be around 63. This particular example isn't random enough.

The experiments have shown how the statistics improve as the number of modules is increased. Here's an early experiment to illustrate this. In this experiment, the bin average was 125, and the SD should be around 11. As the number of modules increases, there's a fairly steady decline of the SD, until it levels out around the expected value.

Table 4: Randomization Results

Module Count	Standard Deviation
10	79.16
20	64.06
30	56.00
40	40.06
50	32.45
60	24.59
70	13.80 <=transition to “random enough”
80	13.38
90	10.30
100	9.97
110	11.61
120	11.36
200	12.30
500	11.08

One important lesson learned with the simulator is that it's important to vary the delays used for the input taps in the E-boxes. In our recent experiments, the delays used are mostly 1-8 bits. But in half the E-boxes, one of the inputs is delayed 15-25 bits; and in 20% of the E-boxes, another input is delayed 35-45 bits. These delays made a big improvement in the measured statistical randomness.

The simulator has also been very useful in evaluating the effectiveness of different choices of the logic functions.

7.5.2. Effectiveness of Various Logic Functions

The results of experiments so far are summarized in the table below. The module count is the (approximate) number of E-boxes with the particular logic function, required to completely randomize the input stream (as measured by the standard deviation of the byte-histogram). This indicates the relative effectiveness of the different logic functions.

At least twice as many modules are required for encryption as are needed for simple randomization.

Table 5: Module Count Requirements

Logic Function	Module Count	Comment
A & (B V C)	*125+	Mixed complements.

AB V CD	*200+	All inputs complemented.
Maj3	25	Alternately 1,2 inputs complemented
Maj5	20	3 inputs complemented
1of3	15	1 input complemented
6:1 mix of A(BvC), Maj3	70-105	

* The top two experiments were run before the delay improvements were added, so these functions aren't as bad as the table indicates.

The experiment suggests that mixing several types of F-function is a good thing to do, but it complicates the construction.

Maj3(A,B,C) is the majority vote of the three inputs. Maj5 is similar, with five inputs. These two functions are relatively simple to build with threshold logic. The most effective mixing function is the 1of3 function. This function is true when exactly one of its three inputs is true. Conceptually, it's a little bit like the exclusive-or function, but it's not possible to define it as a combination of simpler Xor functions.

7.6. Authentication

Any cryptographic message that might be subject to tampering, or even plain transmission errors, must be authenticated. Stream ciphers are no exception. For our system, we will periodically insert a 32-bit CRC into the data stream to be encrypted. The receiver will validate the CRC as part of the decryption step, and flag or drop any bad messages.

8. Conclusion

This document detailed the PSpice modeling efforts for S-SEED logic and demonstrated encapsulated models of XOR, NAND, and NOR functions. Future tasks include simulating feedback loops and more complex logic circuits. A stream cipher for photonic encryption has also been devised and its statistical randomness has been simulated in C language.

9. References

[1] J. D. Tang et al., "Photonic Encryption using All Optical Logic," SAND 2003-4474, December 2003.

[2] S. Okamoto, "Photonic Transport Network Architecture and OA&M Technologies to Create Large-Scale Robust Networks," IEEE Journal on Selected Areas in Communications, vol. 16, no. 7, September, 1998.

[3] D. Blumenthal, "Photonic Packet Switching and Optical Label Swapping," Op. Nets. Mag., Vol. 2, no. 6, November/December 2001, pp. 54-65.

[4] S. Tarek, et. al., "Optical Packet Switching in Core Networks: Between Vision and Reality," IEEE Communications, vol. 40, no. 9, Sepember, 2002.

[5] A.L. Lentine et. al, "Symmetric self-electrooptic effect device; optical set-reset latch, differential logic gate, and differential modulator/detector," IEEE J. Quantum Electronics, vol. 25, no. 8, pp. 1928-1936, August 1989.

[6] D.K. Serkland, I.J. Fritz, T. Sullivan, J.H. Burkhart, and J.F. Klem, "December 2, 1999 Intermediate Progress Report: Switching Speed Measurements of Symmetric Self-Electrooptic Effect Device at 865 nm."

Distribution

MS 0801	A.L Hale, 9300
MS 9003	K.E. Washington, 8900
MS 9011	T.J. Toole, 8941
MS 9012	R.D. Gay, 8949
MS 9011	B.V. Hess, 8941
MS 9019	S. Marburger, 89451
MS 9012	S.C. Gray, 8949
MS 9012	M.G. Mitchell, 8949
MS 0801	M.R. Sjulin, 9330
MS 0801	W.F. Mason, 9320
MS 0801	D.S. Rarick, 9310
MS 0806	G.K. Rogers, 9312
MS 0813	R.M. Cahoon, 9311
MS 0795	P.C. Jones, 9317
MS 0799	G.E. Connor, 9333
MS 0788	P.L. Manke, 9338
MS 0812	M.J. Benson, 9334
MS 0806	Len Stans, 9336
MS 0806	J.P. Brenkosh, 9336
MS 0806	J.M. Eldridge, 9336
MS 0806	A. Ganti, 9336
MS 0806	S.A. Gossage, 9336
MS 0806	T.C. Hu, 9336
MS 0806	B.R. Kellogg, 9336
MS 0806	L.G. Martinez, 9336
MS 0806	M.M. Miller, 9336
MS 0806	J.H. Naegle, 9336
MS 0806	R.R. Olsberg, 9336
MS 0806	L.G. Pierson, 9336
MS 0806	T.J. Pratt, 9336
MS 0806	J.A. Schutt, 9336
MS 0806	J.D. Tang, 9336
MS 0806	L.F. Tolendino, 9336
MS 0806	J.S. Wertz, 9336
MS 0806	D.J. Wiener, 9336
MS 0806	E.L. Witzke, 9336
MS 0603	C.T. Sullivan, 1742
MS 0603	D.K. Serkland, 1742
MS 0603	G.A. Vawter, 1742
MS 0603	J. Guo, 1742
MS 0785	T.S. McDonald, 5516
MS 0785	R.C. Schroeppel, 5516
MS 9019	Central Technical Files, 8945-1
MS 0899	Technical Library, 9616 (2)
MS 0874	D.W. Palmer, 1751
MS 0874	P.J. Robertson, 1751
MS 0874	K.L. Gass, 1751
MS 1206	J.V. Vonderheide, 5942
MS 1202	A.N. Campbell, 5940