GPIO DESIGN, LAYOUT, SIMULATION AND
ESD CLAMP PLACEMENT CALCULATOR

by

Shiju Abraham

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in partial fulfillment

of the requirements

for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2014

Acknowledgements

I would like to thank the Lord Almighty for giving me the opportunity to do MS by Thesis and a co-op with GLOBALFOUNDRIES. I thank God for giving me the ability, patience, perseverance and good health to get this work done.

I would like to acknowledge the support and guidance from my advisor Dr. W. Alan Davis. He has been patient and given continuous feedback on my work. It was a privilege to work under his supervision. I express my immense gratitude to Dr. Davis.

I would like to thank Dr. Jonathan Bredow and Dr. Kambiz Alavi for being members of my thesis committee and giving me valuable feedback. I thank office assistants (Ms. Ann, Ms. Janice, and Ms. Pauline) of the EE department at UT Arlington.

I thank Dr. Mahadeva Iyer Natarajan (Natarajan), Mr. Anil Kumar, Mr. Manjunatha Prabhu and Mr. Chien-Hsin Lee (Ano) for the constant guidance and support they provided while I did my co-op with GLOBALFOUNDRIES and worked on a similar topic. They have been patient and attentive throughout the co-op and reviewed my work at each phase.

Finally and most importantly, the thesis would not have been possible without the love and support of my parents (Mr. P.A. Abraham, Ms. P.M. Mariamma), other immediate family members and friends (Mr. Sujith S. Dermal, Ms. Stella Kurian, Ms. Mitha Ann Philip). I thank them from the bottom of my heart for the encouragement and support they provided.

June 30, 2014

Abstract

GPIO DESIGN, LAYOUT, SIMULATION AND ESD CLAMP PLACEMENT CALCULATOR

Shiju Abraham, M.S.

The University of Texas at Arlington, 2014

Supervising Professor: Dr. W. Alan Davis

Input/Output (IO) circuits enable interface between logic circuitry and the actual or raw information to be processed. They also help to isolate the integrated circuit from the unsafe, unknown and noisy environment. IOs come in many flavors and the General Purpose IO (GPIO) is one among them. GPIOs can operate as an input, output or a bi-directional circuit. The purpose of this work is to design an area optimized industrial quality bi-directional GPIO with separate enable signal for transmitter and receiver which can drive a current of at least 16 mA into the PAD (the circuit point where the capacitive load is connected). The typical IO power level (the power at which a Printed Circuit Board, PCB operates) is 1.8 V and the core (the logic circuitry) power level is 1.0 V. Drive strength control and slew rate control are included in the GPIO implementation. Since many GPIOs could be placed in an IO ring (IOs placed around the periphery of the chip), its placement optimization is important for optimal chip area, as well as, robust IO ring from performance and qualification requirements.

IOs need to be protected from ESD events. One of the key ESD protection methodologies involve accurate ESD device sizing versus ESD current path distance optimization. A calculator is developed to predict the optimum distance at which a power clamp should be present for a given IO ESD device size and overall current carrying element

availability. This tool is supposed to get certain inputs regarding the ESD protection devices from the user and suggest an optimum distance at which a power clamp should be placed in an IO ring.

This work is intended to produce one of the most compact GPIOs in the given technology node (the distance between source and drain of the CMOS transistor), 28 nm and a clamp placement calculator which works for different technology nodes.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction to IOs

Input/Output (IO) circuits enable a chip to communicate with the external world. They are placed at the periphery of a chip and provide an interface between the chip and the external world. As the internal circuitry grows in speed and efficiency, it processes data faster. The number of bits per word has been increasing. Chips communicate with the external circuitry for storage, display or further processing of the processed data. Faster chips and longer word size combine to represent a higher bandwidth requirement for interface circuitry. Matching IO circuits, in terms of speed and bandwidth, are critical to make sure that the processing power and efficiency of the internal circuitry or the core circuitry is best used. A typical chip plan will have IOs situated along the periphery of the chip as shown in Figure 1-1.

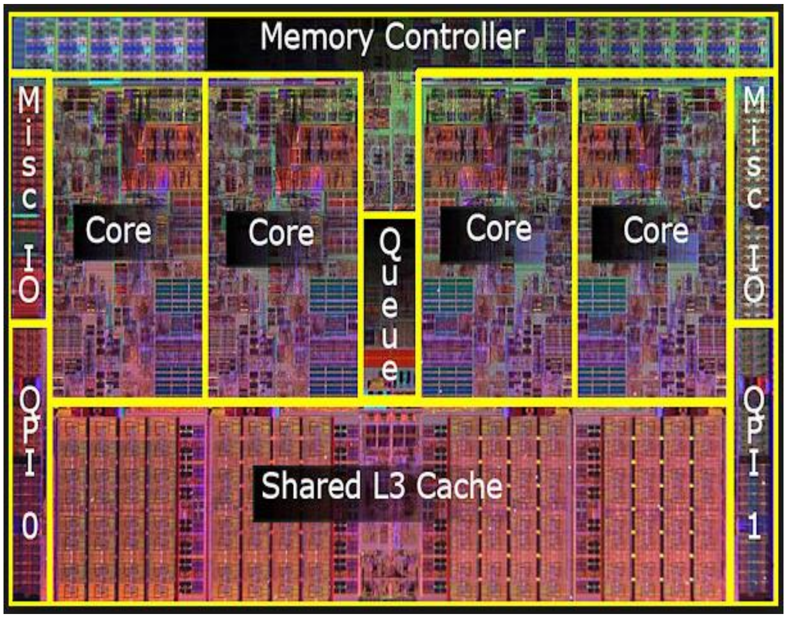


Figure 1-1 Intel i7 chip layout. Source: Anandtech

The electrical signal outside the chip is unknown and possibly unsafe for the internal circuitry. IOs help isolate the chip from such an environment and helps convert the external signal to a form where the internal circuit can process it. This is in the form of voltage level conversion, from a board level voltage, of say 3 V, to a chip level voltage, of say 1.1 V, by improving the eye diagram of the noise affected signal as shown in **Error! Reference source not found.** in a circuit arrangement as shown in **Error! Reference source not found.**.

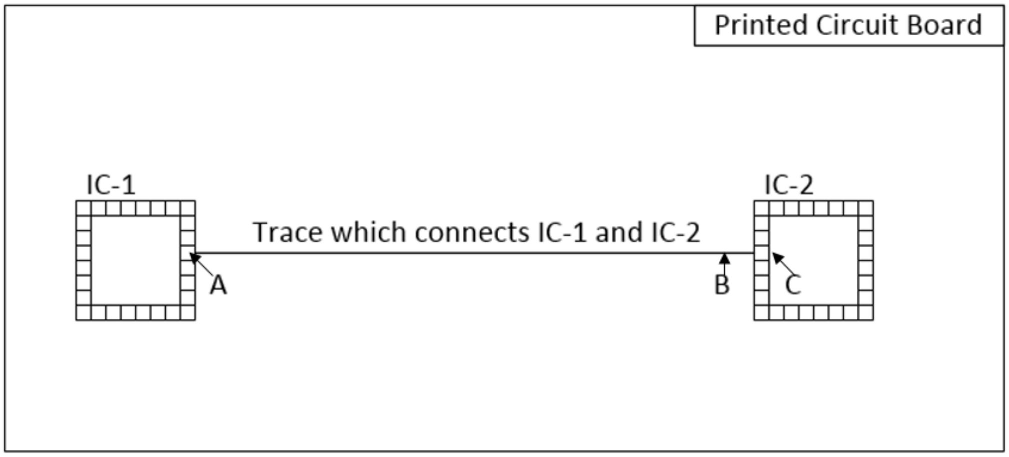
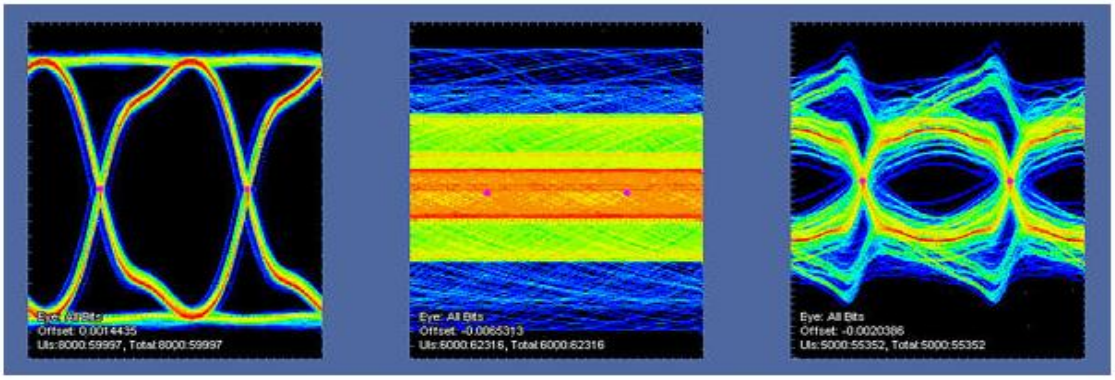


Figure 1-2 Two integrated circuit chips communicating with each other on a PCB



(a) (b) (c)

Figure 1-3 Signal at different stages: (a) Signal at transmitter (Point A in Figure 1-2), (b) Signal on the circuit board (Point B in Figure 1-2), (c) Signal at receiver after conditioning (Point C in Figure 1-2). Source: http://www.tek.com/datasheet/sdla.

As part of this thesis, development (design, layout and simulation) of a General Purpose Input/Output (GPIO) buffer is done with an intention of achieving a compact layout compared to the existing or already achieved layout area. This thesis describes a spreadsheet calculator that determines placement of Electro Static Discharge (ESD) clamp placement which works based on the layout information provided by the user and the technology specific information forms the other part of the thesis. This thesis report is organized into six chapters.

Chapter 1 presents the basic information related to Input/Output buffers like signaling standards, types of IOs, physical arrangement of IOs in an integrated circuit (IC) chip and performance metrics for an IO. This creates the basic understanding required to perform design of IOs.

Chapter 2 explains the General Purpose Input/Output (GPIO) buffer development. It starts from specification and goes through design, layout and simulation of the GPIO.

Chapter 3 provides an introduction to ESD events in IC chips. Various test models and protection schemes are explained along with RC-triggered power clamp circuit.

Chapter 4 presents ESD network analysis which is eventually used by the clamp placement calculator, “IO planner”. The calculator is an outcome of the thesis work. The chapter gives a clear idea about the different circuit components present in an ESD network and how the analysis is done.

Chapter 5 explains the method used for of the clamp placement in the calculator tool. It clearly explains how each component of the tool functions and brings about the final result of finding the optimum placement for the clamps in an IO ring.

Chapter 6 concludes the thesis report by explaining the merits and future scope of the work.

1.1 Signaling Standards

There are two basic forms of IO signaling: single-ended signaling and differential signaling. In single-ended signaling, one wire usually carries a varying voltage that represents the signal, while the other wire is connected to a reference voltage, usually ground as shown in

Figure 1-4. Figure 1-4 shows two singled-ended signals. This type of signaling is less expensive to implement, but it lacks the ability to reject noise introduced during transmission or reception.

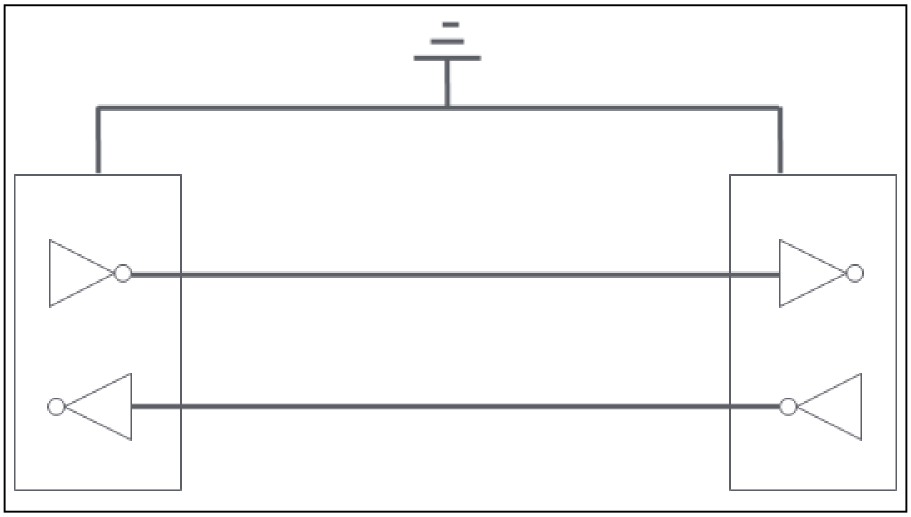


Figure 1-4 Representation of single-ended IO signaling

In differential signaling, complementary signals are sent on a pair of wires and the voltage on one is measured with reference to the voltage at the other wire. Differential signaling is more complex and expensive to implement, but it has the ability to reject common-mode noise. A general representation of differential signaling is given in Figure 1-5.

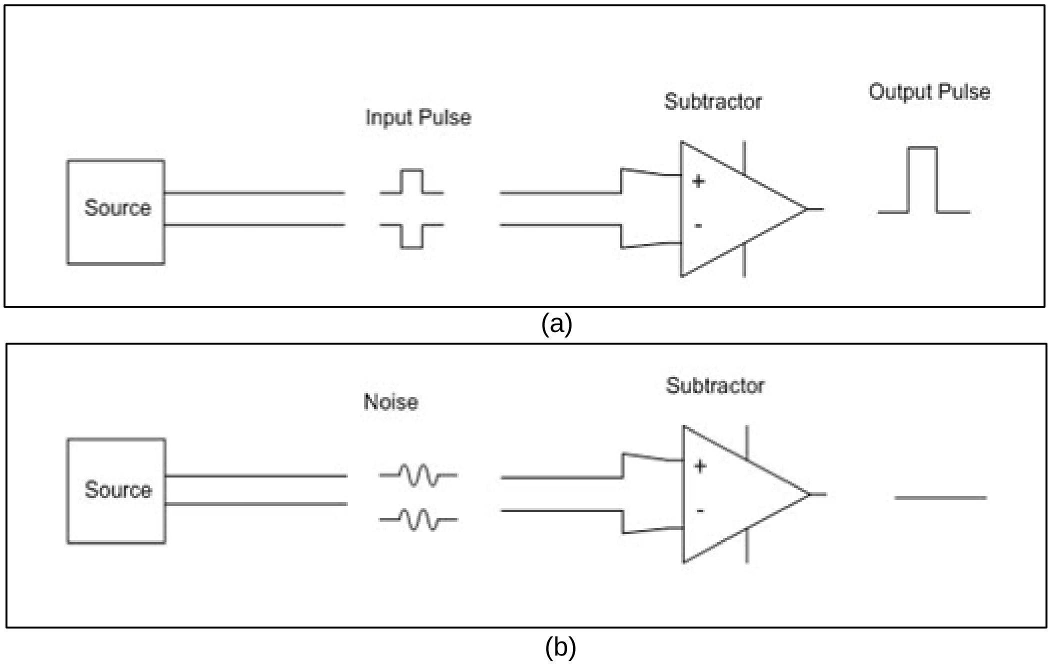


Figure 1-5 Representation of differential IO signaling: (a) Differential signaling, (b) Common-mode noise rejection

1.2 Types of IOs

Depending on the type of application, IOs can be classified into different types. Some of them are,

- Input
- Output (2-state or 3-state)
- Bi-directional
- Open-drain
- Low Voltage Differential Signaling (LVDS)

Each of these will be described in the following sections. The term ‘buffer’ is used alternately for “IO” since IOs does not perform any logic operation on the signals. Convention for transistor symbols used is explained in section 1.6.

1.2.1 Input Buffer

The input buffer passes external data to the core. It performs the level conversion from the external voltage to the core voltage level. It helps improve the signal by performing some kind of signal conditioning. ESD diodes associated with the input buffer help protect Integrated Circuit (IC) chips from damage due to ESD events. A generic representation of an input buffer is shown in Figure 1-6.

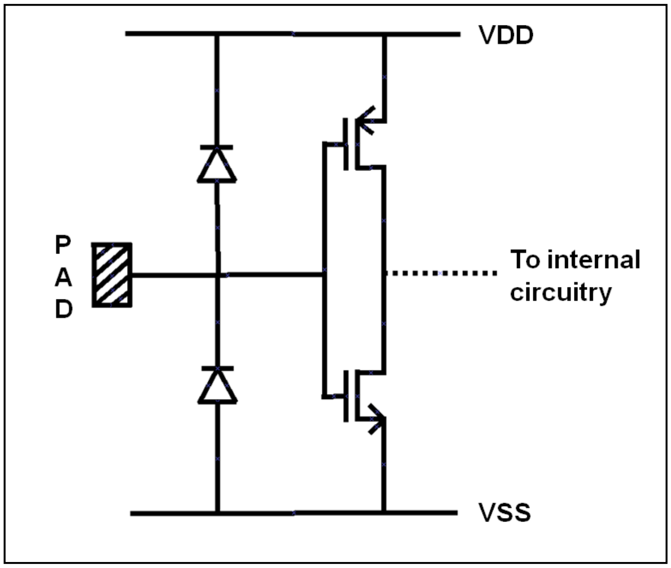


Figure 1-6 Input buffer

1.2.2 Output Buffer

The output buffer passes data from the core to the external world which is usually another component on the Printed Circuit Board (PCB) through a track. It performs level conversion from the core level voltage to the IO level output voltage (the motherboard voltage level). Output buffers can be either 2-state or 3-state depending on the application. For a 3-state buffer, the three states are logic low, logic high and high impedance. A 3-state buffer will have an enable signal which facilitates achieving high impedance (Hi-Z) at the PAD (designated as PAD in the circuit layout). ESD diodes associated with the output buffer also help protect ICs from damage due to ESD events. A generic representation of an output buffer is shown in Figure 1-7.

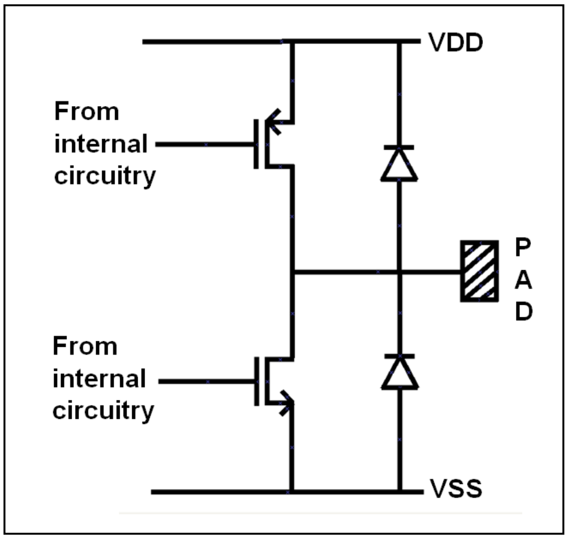


Figure 1-7 3-state output buffer

1.2.3 Bi-directional Buffer

A bi-directional buffer functions as both an input and an output buffer. The enable signal which comes from the core determines if the buffer needs to be configured as an input buffer or an output buffer. It is designed such that when enabled as an input buffer, the PAD is at a high impedance state. There can be designs where both an input and an output buffer have separate enable signals. A generic representation of a bi-directional buffer is shown in Figure 1-8.

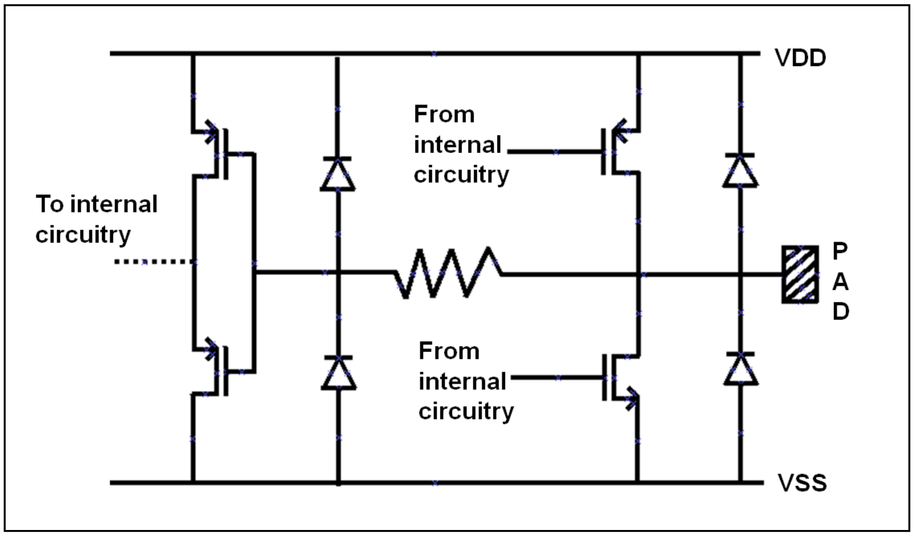


Figure 1-8 Bi-directional IO buffer

1.2.4 Open-drain

Open-drain buffers come with pull-up resistors instead of a PMOS transistor. This pull-up resistor is external to the chip mostly and needs to be connected to the specified termination voltage (VTT). Since VTT determines the high-level output voltage value, open-drain buffers are quite often used in voltage translation applications. A generic representation of an open-drain buffer is shown in Figure 1-9.

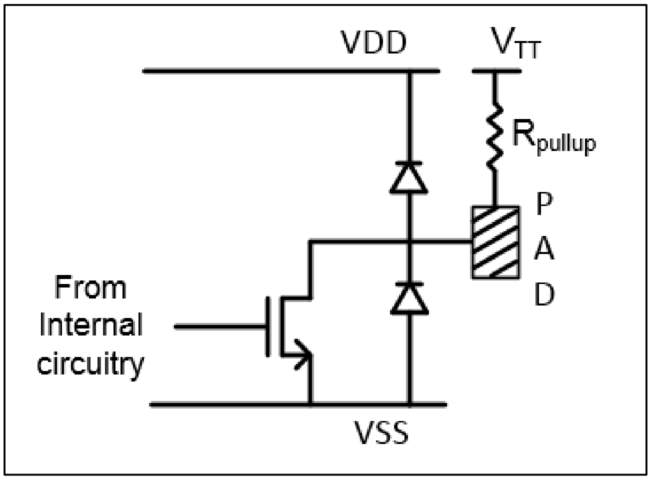


Figure 1-9 Open-drain buffer

1.2.5 LVDS

Differential buffers can also be used depending on the application. LVDS buffers help achieve higher speed, lower power dissipation, and common-mode noise rejection compared to a single-ended buffer. But, these advantages come at the design cost of time and money. There are receiver and driver designs based on LVDS technology. A generic representation of an LVDS driver is shown in Figure 1-10.

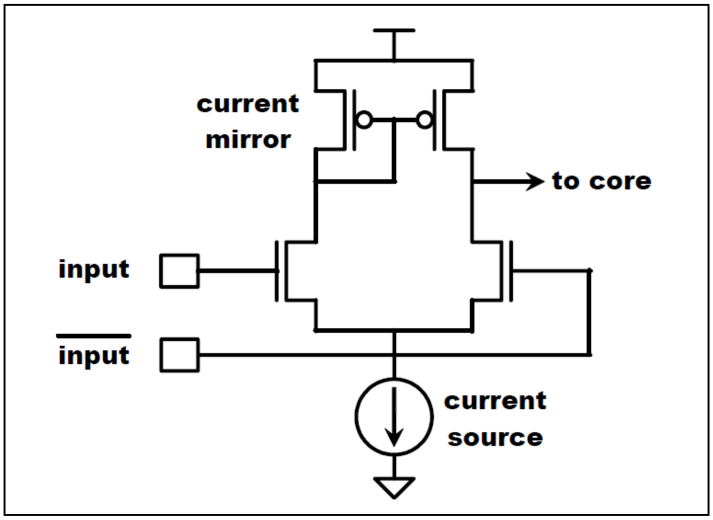


Figure 1-10 LVDS driver

1.3 Physical Arrangement of IOs

Based on the application for which the chip is designed, it is classified as a pad-limited design or a core-limited design. A pad-limited design, shown in Figure 1-11, occurs when the die area is determined by the area occupied by IOs in the design. IOs in such designs are tall and skinny (geometrically narrow) making room for a large number of IOs.

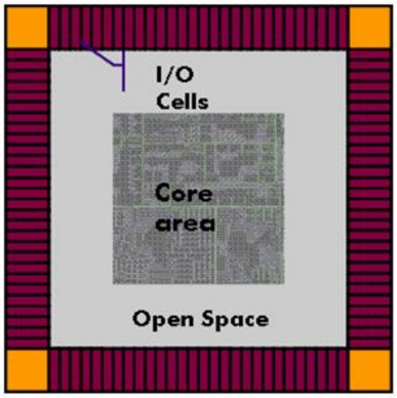


Figure 1-11 Pad-limited design. Source: http://www.chipdesignmag.com.

A core-limited design, shown in Figure 1-12, occurs when the die area is determined by the area occupied by core logic in the design. IOs in such designs are short and fat (geometrically wide) making more room for the core circuitry.

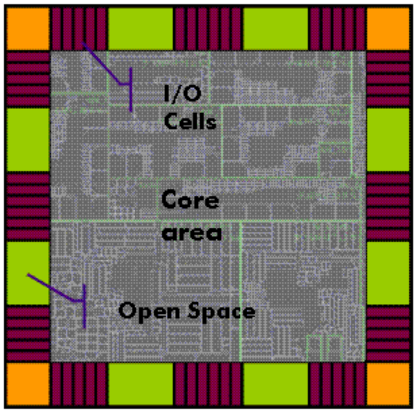


Figure 1-12 Core-limited design. Source: http://www.chipdesignmag.com.

An IO can be inline or staggered depending on the physical arrangement, as shown in Figure 1-13 and Figure 1-14.

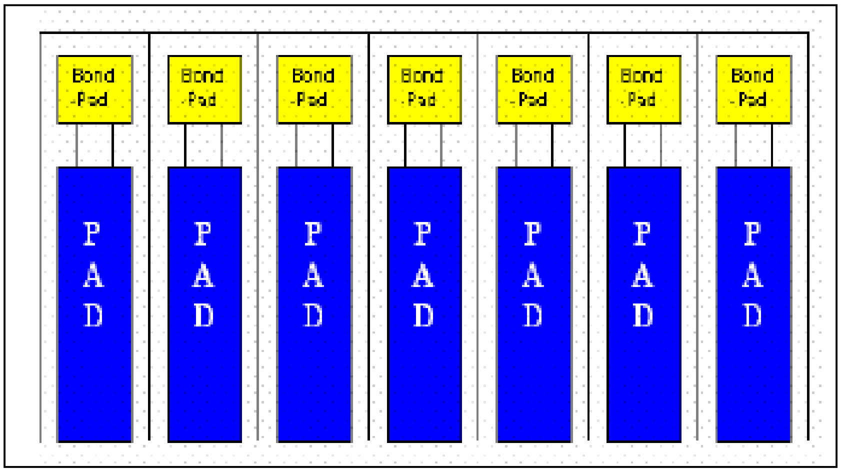


Figure 1-13 Inline arrangement [1]

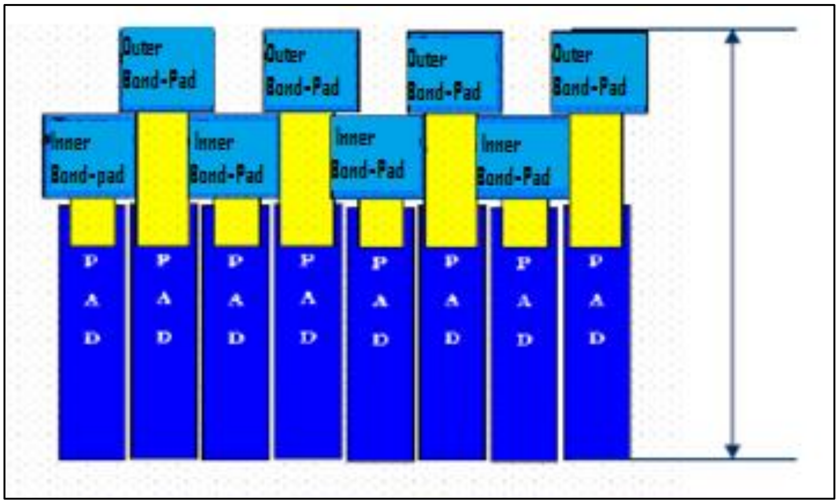


Figure 1-14 Staggered arrangement [1]

IOs normally form a ring around the core circuitry, along the periphery of the chip. The main components of the IO-ring are signal IOs, power IOs, Fill cells, and corner cells

A signal IO can either be a digital IO or an analog IO; it can be of any of the type explained above depending on the type of application. On-chip ESD protection circuits are part of the signal IOs. A General Purpose IO (GPIO) is an example of a signal IO.

A power IO functions to provide a power supply for the chip. Power clamps which are part of the IOs, help reduce risks due to ESD events for the chip. A rule of thumb is to have as much power IOs, especially for the core power supply, as possible so that the chip is supplied with stable and robust power.

Fill IOs are used to fill the gaps after placing signal and power IOs. Sometimes, the number of IOs will be too small to fill the area available for an IO ring or the length of an IO ring may not be a whole number multiple of the geometric width of the signal and power IO. In both these cases, FILL cells are used to fill the gap. There need not be any active device in these cells. However decoupling capacitors (decaps) – capacitors connected between power supply and ground which act as charge reservoirs when power supply fluctuates due to reasons such as simultaneous switching outputs – can be included in FILL cells to make use of the layout area.

It is important to have continuity of power rails around the chip. Corner cells help to maintain the power continuity at the four corners of the chip. Corner cells do not have any device in them. They have only the horizontal power routes to ensure continuity throughout the whole chip. A corner cell is shown in Figure 1-15.

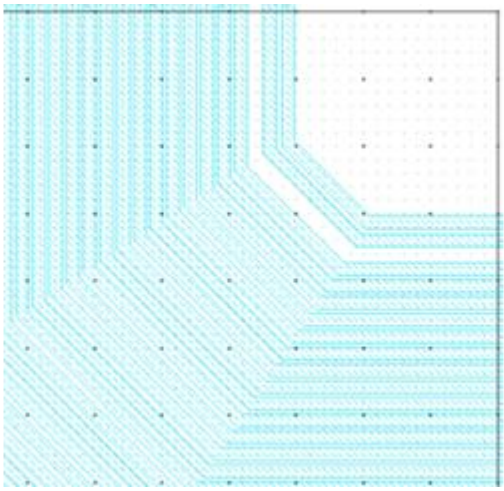


Figure 1-15 Corner cell with highlighted cell boundary and M5 metal shown

This work involved creating GPIOs, Power IOs (VDDIO, VDDC, VGND (VSS)) and fill IO (FILL1, FILL2, FILL3, FILL5, FILL10) cells. The corner cell is not part of the current work since the intention was to create an IO bank (which sits on periphery of the chip along one side) which would fit on one side of the chip.

1.4 Performance Metrics for IOs

It is important to understand some of the performance criteria of IOs in order to understand the technical specification for the design and to perform design and development of the IO circuit. Basic performance criteria for an IO are defined below [2].

IO-level supply voltage (VDDIO): This specifies the voltage that can be safely applied to any VDDIO pin. Exceeding the specified voltage level may harm the chip. The level shifter inside the IO is designed to perform voltage conversion from VDDIO to the core level voltage.

Core-level supply voltage (VDDC): This specifies the power supply voltage that can be safely applied to a VDDC pin.

Low-level input voltage (VIL): This is the maximum input voltage which is recognized as a logic-LOW by the IO circuit.

High-level input voltage (VIH): The minimum input voltage which is recognized as a logic-HIGH by the IO circuit.

Low-level output voltage (VOL): The voltage level at an output terminal of the IO when the input is within certain specification, produces a specified low level at the output of the IO.

High-level output voltage (VOH): The voltage level at an output terminal with input conditions applied that, according to the product specifications, will establish a high level at the output.

Positive-going input threshold level (VT+): The input threshold voltage when the input is rising.

Positive-going input threshold level (VT-): The input threshold voltage when the input is falling.

Hysteresis (ΔVT): Hysteresis is the difference between positive-going and negative-going input threshold voltages as shown in Figure 1-16.

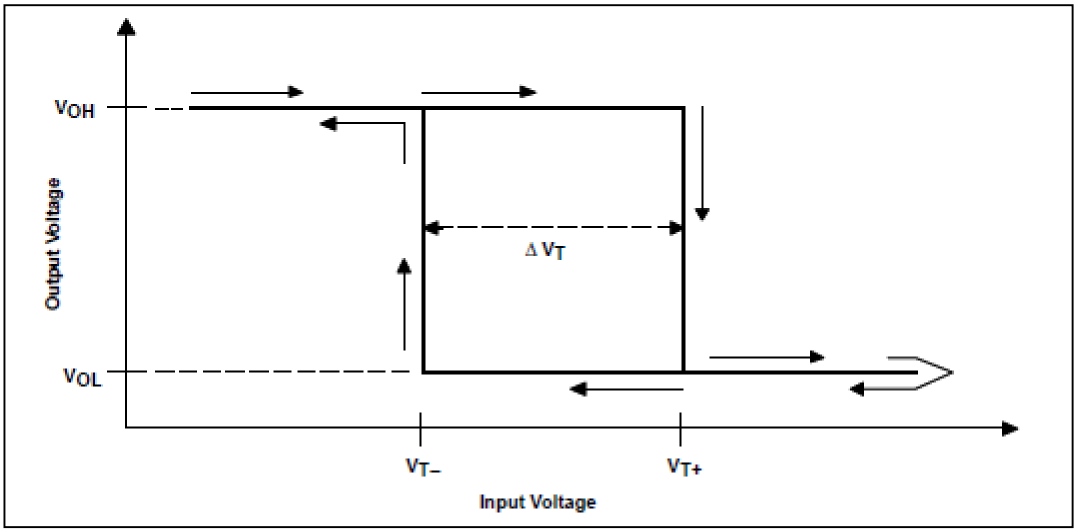


Figure 1-16 Hysteresis [2]

Termination voltage (VTT): A supply voltage used to terminate a bus (most commonly used in open-drain devices) and in generating a reference voltage for differential buffers.

Low-level input current (IIL): The current into an input terminal when a specified low-level voltage is applied to that input.

High-level input current (IIH): The current into an input terminal when a specified high-level voltage is applied to that input.

Low-level output current (IOL): The current into the output terminal with input conditions (input signal thresholds, slew rate, noise level) applied that, according to the product specification, will establish a low level at the output.

High-level output current (IOH): The current into the output terminal with input conditions applied that, according to the product specification, establishes a high level at the output.

Operating temperature (T): It is the range of temperature in which the IO is meant to be functional.

Frequency of operation (Fmax): The highest frequency at which the circuit can be driven while maintaining proper operation.

Duty cycle: In general, it represents the time for which the circuit/device is operated. In square wave terms, it is the ratio of time duration for which the circuit is ON (logic-HIGH) to the total time. Duty cycle = t_{ON} / (t_{ON} + t_{OFF}) x 100.

Propagation delay, high-level to low-level output (tpHL): The time interval between specified reference points on the input and output voltage waveforms with the output changing from the defined high level to the defined low level as shown in Figure 1-17.

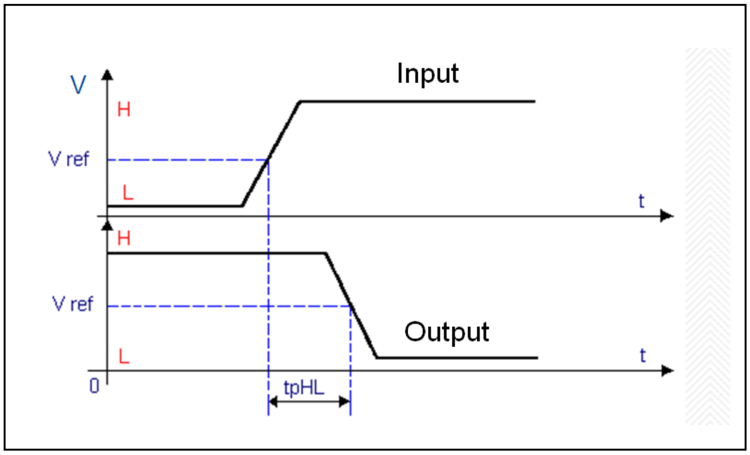


Figure 1-17 Measurement of propagation delay, tpHL

Propagation delay, low-level to high-level output (tpLH): The time interval between specified reference points on the input and output voltage waveforms with the output changing from the defined low level to the defined high level as shown in Figure 1-18.

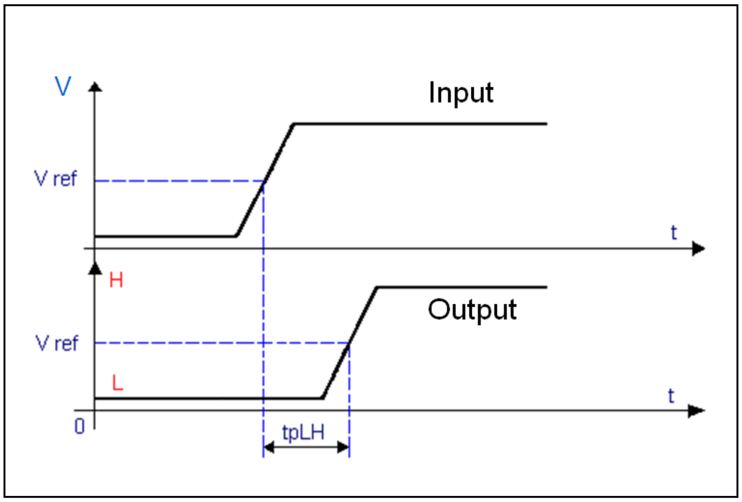


Figure 1-18 Measurement of propagation delay, tpLH

Fall time (tf): This is the time taken for the output signal to change from one reference value to another reference value, while changing from the defined high level to the defined low level (typically 90% to 10% of the final output).

Rise time (tr): This is the time taken for the output signal to change from one reference value to another reference value, while changing from the defined low level to the defined high level (typically 10% to 90% of the final output).

Drive strength: This is the amount of current that can be drawn from the IO buffer while maintaining the appropriate output voltage levels for corresponding logic level inputs.

1.5 Schematic, Layout, Simulation Standards

The conventions used in the schematic entry and in the layout of the design are explained below:

Pitch of an IO: Pitch is the spacing from a point on one PAD to the same point on an adjacent PAD.

Slew rate of a signal: Rate of change of voltage (dV/dt) is defined as the slew rate of a signal. It is the transition time of the signal.

NMOS transistor: Both the symbols shown in Figure 1-19 represent an NMOS transistor.

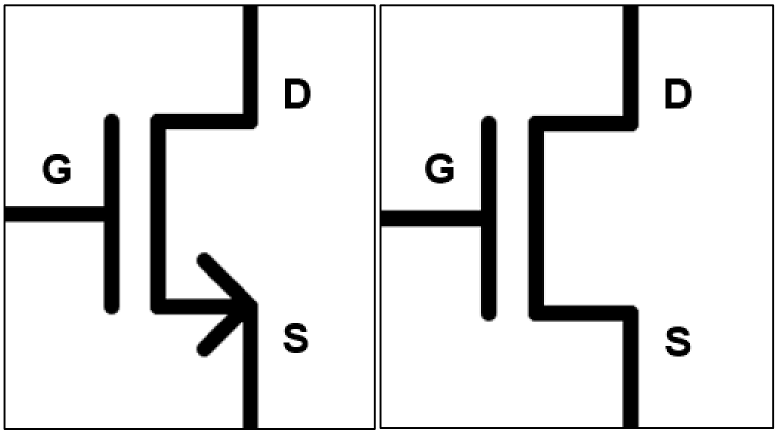


Figure 1-19 Symbol of an NMOS transistor

PMOS transistor: Both the symbols shown in Figure 1-20 represent a PMOS transistor.

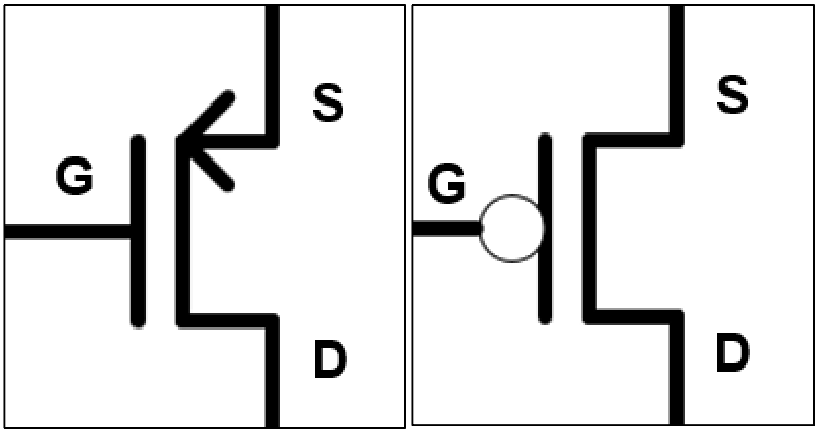


Figure 1-20 Symbol of a PMOS transistor

Generally, for a Complementary Metal Oxide Semiconductor (CMOS) transistor, source (S) and drain (D) terminals are interchangeable due to the physical structure it has. For a PMOS transistor, source is the terminal which is at a higher potential among the two terminals (source and drain). Similarly, for an NMOS transistor, source is the terminal which is at the lower potential among the two terminals. The terminal marked “G” represents gate of the transistor.

Fingers of a transistor: A long transistor can be split into several smaller transistors that are hooked up in parallel. This will enable diffusion sharing and reduce parasitic resistance due to parallel combination. In Figure 1-21, a transistor with a larger width is shown split into smaller transistors.

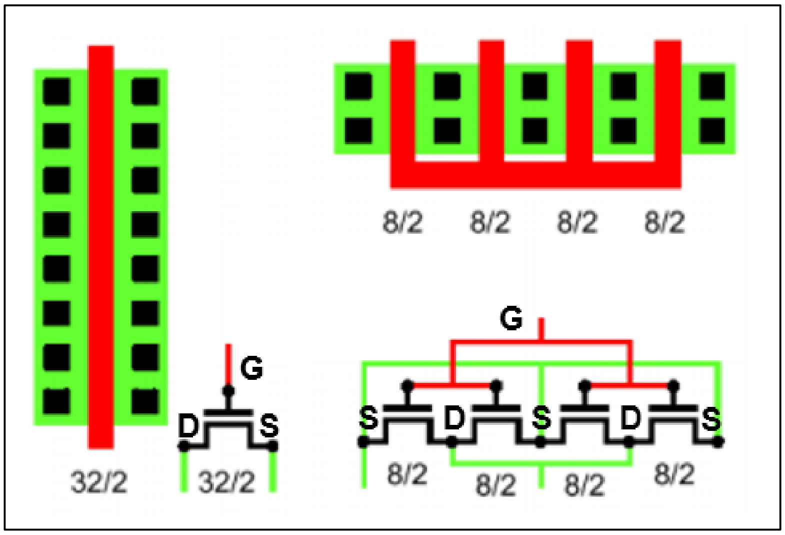


Figure 1-21 Fingers of a transistor [3]

Buffer: A buffer is a circuit arrangement which passes the input signal without any logic change to the output. Two inverters in series is the simplest form of a buffer circuit. The buffer circuit can retrieve the shape of a signal if it has deteriorated. Also, a buffer can be used to add delay in a circuit. A “core buffer” in this document refers to a buffer circuit which buffers a signal from core.

Process Voltage Temperature (PVT) corner: PVT corners model variations in process, voltage and temperature. PVTs model inter-chip variations. The process can be a slow (min), typical or a fast (max) process. The voltage will have a typical value which is the ideal voltage for the circuit to function as intended. Minimum and maximum values can be 90% and 110% of the typical voltage values. The typical temperature is the room temperature (25 °C). The min value is -25 °C and the max value is 125 °C for temperature. The PVT values can be set according to the application of the IC. A circuit is simulated for different combinations of process, voltage and temperature values in order to ensure proper functioning of the circuit in silicon.

Parasitic extraction: A real wire (metal route) has resistance, capacitance and inductance. Parasitic extraction is the process of finding the equivalent resistance, capacitance and inductance values of interconnects used in the IC layout. Software tools are available to perform parasitic extraction.

Layout parameters of a transistor: The length (L) of a transistor is the distance between it's source and drain. The width (W) of a transistor refers to how wide the conduction channel is. The wider the channel, the higher the current capability. The length and width parameters for a PMOS transistor are illustrated in Figure 1-22.

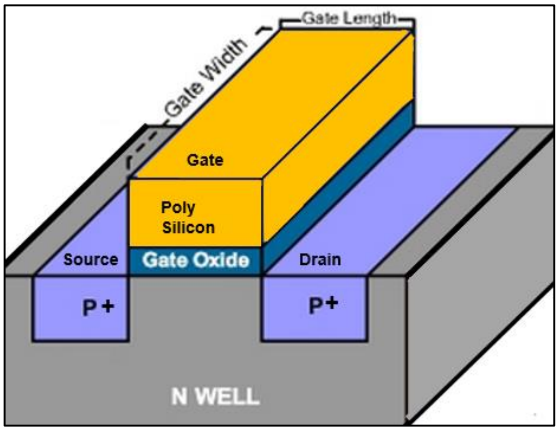


Figure 1-22 Length and width of a transistor

Process Development Kit (PDK): A process development kit refers to a set of files which is used to model devices for a certain technology for a certain foundry.

SG device: This is a PDK specific term used in the spreadsheet and in this document. This represents CMOS devices with a particular minimum gate length and these devices are used for core circuits.

EG device: This is a PDK specific term used in the spreadsheet and in this document. This represents CMOS devices with a higher minimum gate length than the “SG” devices. EG devices are used for IO and clamp circuits mainly.

Layout parameters of a diode: Length of the diode refers to the length of the device used to construct the diode. The perimeter refers to the distance around the PN junction formed. For example, the perimeter of the diode shown in Figure 1-23 is equal to 4 x W.

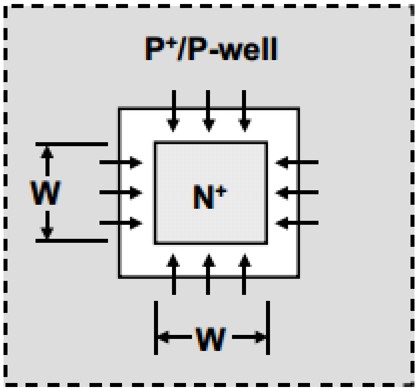


Figure 1-23 Perimeter of a diode: An STI-bound N+/PW diode. Arrows highlight the primary current conduction path [4]

Shallow Trench Isolation diode: Shallow Trench Isolation (STI) is a process typically used at the semiconductor surface to create isolation structures. The cross sectional view of an STI diode is shown in Figure 1-24.

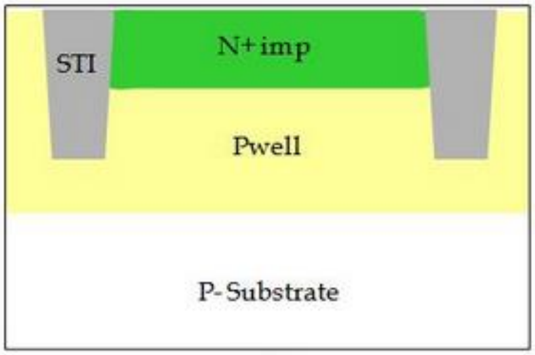


Figure 1-24 STI diode – N+/PW type. Source: http://www.intechopen.com/ .

Polysilicon diode: The polysilicon diode is another diode which is created in the polysilicon layer instead of the silicon substrate. [5]

Stacked devices: In high frequency applications, instead of using one forward biased diode, two or more diodes are used in series. If all stacked diodes are the same, the parasitic resistance of the diode structure is multiplied by the number of diodes in series and the capacitance is divided by the same number. The “STI string” diode mentioned in the spreadsheet and in this document refers to stacked devices. [5]

P+/NWell, N+/PWell (P+/NW, N+/PW) diodes: Both STI and poly diodes can be P+/NW or N+/PW type depending on the physical layers used and their doping profiles. If it is an N+/PW type, the structure will have a P-type well which forms the anode and an N+ implant which forms the cathode. For a P+/NW type, P+ implant forms the anode and N-well forms the cathode.

Chapter 2

GPIO Development

The design a GPIO according to a set of specifications requires circuit design, layout and SPICE simulation or validation. The end goal is to create an IO bank of 2 mm x 2 mm size including the GPIO and other relevant cells (power cells and fill cells) for the IO bank.

2.1 Specification

Specification for the GPIO is given in Table 2-1.

Table 2-1 Target specifications of the GPIO

Parameter	Signal levels			Unit
	Min	Typical	Max	
VDDIO	1.62	1.8	1.98	V
VDDC	0.9	1.0	1.1	V
Temperature	-25	50	125	°C
Frequency		200		MHz
Duty Cycle	45	50	55	
VIL			0.3 x VDDIO	V
VIH	0.7 x VDDIO			V
VOL			0.2	V
VOH	VDDIO – 0.2			V
IOH	16			mA
VHYS	300			mV

The transmitter and the receiver blocks of the GPIO have separate active-HIGH enable signals which are represented by the pins TX_EN and RX_EN respectively. When TX_EN = HIGH, the GPIO passes a signal from the core circuitry to the PAD. Similarly, When RX_EN = HIGH, the GPIO passes signal from the PAD to the core circuitry. The truth table of the GPIO is provided in Table 2-2 and Table 2-3. In this document, the word “transmitter” refers to the output buffer and the word “receiver” refers to the input buffer. The driver is a circuit element that can push a definite amount of current through the load connected.

Table 2-2 Truth table - Transmitter

Transmitter		
DATA_from_CORE	TX_EN	PAD
X (don’t care)	0	Hi-Z (high impedance)
0	1	0
1	1	1

Table 2-3 Truth table - Receiver

Receiver		
PAD	RX_EN	DATA_to_CORE
X (don’t care)	0	Hi-Z (high impedance)
0	1	0
1	1	1

2.2 General Block Diagram of a Bi-directional GPIO

The input signal for the transmitter block comes from the core circuit. A core buffer is required since the signal may come through a long route from the core and may be distorted. A level shifter in the IO converts the VDDC level to the VDDIO level. The PAD needs to be at high impedance state while the GPIO functions as an input buffer. If the same signal controls both P-driver and N-driver gates, then the PAD will be either at logic high or at logic low. So, a tri-state machine is required here that generates separate signals to control P-driver and N-driver. The P-driver and the N-driver constitutes the main driver block. Since the main drivers may be huge, a pre-driver stage is also required. The block diagram representation of a transmitter circuit is shown in Figure 2-1.

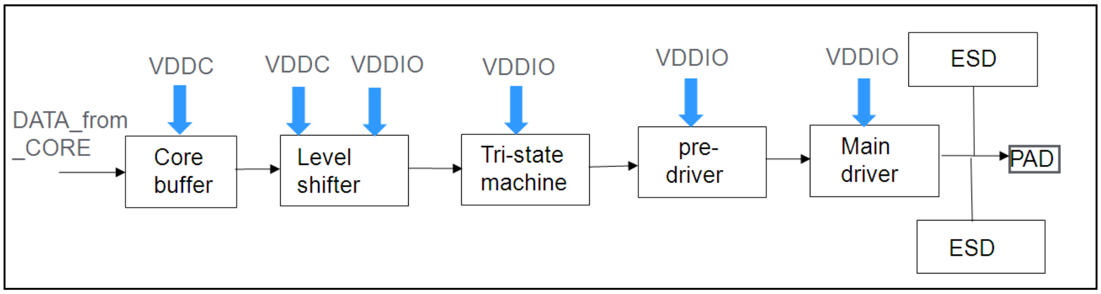


Figure 2-1 Output buffer - block diagram

The receiver block contains an input stage which includes circuitry for hysteresis control. As shown in Figure 1-16, hysteresis control ensures that the receiver circuit is not susceptible to noise to the designed extent. For this work, the hysteresis voltage, ΔVT = 300 mV. The level shifter block changes the voltage-level from the VDDIO level to the VDDC level, then the data signal is passed to the core through a core buffer. A core buffer is required since the signal may have to be routed a long distance to reach the core circuit which the receiver is driving. The ESD protection diodes are connected to the PAD. The block diagram representation of a receiver circuit is shown in Figure 2-2.

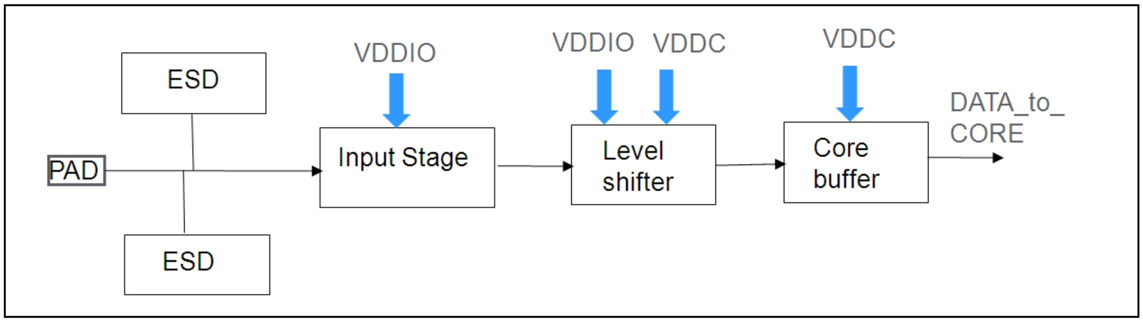


Figure 2-2 Input buffer - block diagram

2.3 Design

Circuit design is performed according to the target specification provided, using the Cadence tool environment and the HSPICE circuit simulator. The GPIO is designed such that it has the following capabilities: drive strength control, crowbar current – current flowing from power supply to ground when both PMOS and NMOS transistors are turned on simultaneously – control, slew rate control for the pre-driver, and the ability to bring high impedance state at PAD.

Although the GPIO is designed for 16 mA drive strength, there may be applications where it does not require 16 mA to drive the load. The 2-bit drive strength control helps to configure the drive strength for 16 mA, 12 mA, 8 mA and 4 mA as required and thereby implements a power management system.

Crowbar current is the short circuit current in a driver circuit caused when both PMOS and NMOS transistors are partially ON during the time the output node changes states. This is similar to the crowbar current present in core logic circuits. The magnitude of current is higher in the case of IOs compared to the logic cells which are part of core circuitry and hence it is desirable to have options to control the crowbar current.

While multiple drivers switch simultaneously and draw current from the same power supply, it causes Simultaneous Switching Noise (SSN). SSN has become more critical due to recent advancements including higher slew rate, reduced power supply level, reduced threshold levels and a larger number of output pins. By having slew rate control for the pre-driver, SSN can be reduced. If the application does not require a higher slew rate at the output, the IO can be operated at a lower slew rate mode and thus reducing the possible SSN. There exists a trade-off between noise sensitivity, slew rate and propagation delay.

By having separate control for the P-driver (pull-up device) and the N-driver (pull-down device) as shown in Figure 2-3, a high impedance can be seen at the PAD. This brings multiple advantages. First, the same PAD can be used as an input as well as an output. Secondly, if multiple drivers are driving the same bus and if some of the drivers are not required, the output of those drivers can be brought to high impedance state so that the bus is not loaded by them.

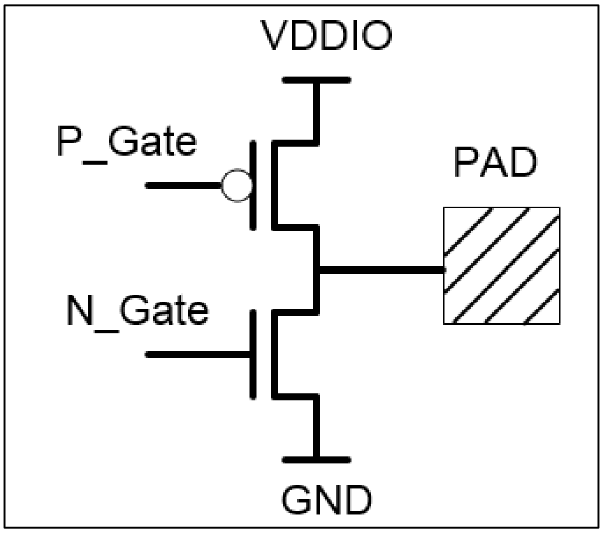


Figure 2-3 Controls for P-driver and N-driver

A top-level schematic of the driver part of the GPIO is shown in Figure 2-4. TX_EN, DATA, d0, d1 are the signals coming from the CORE and they go through the level shifter blocks to be converted to the VDDIO level signal. The level shifted signals act as input to the tri-state machine circuit to generate separate controls for P-driver and N-driver. The output of tri-state machine circuit go through the pre-driver circuit and drive the P-driver and N-driver.

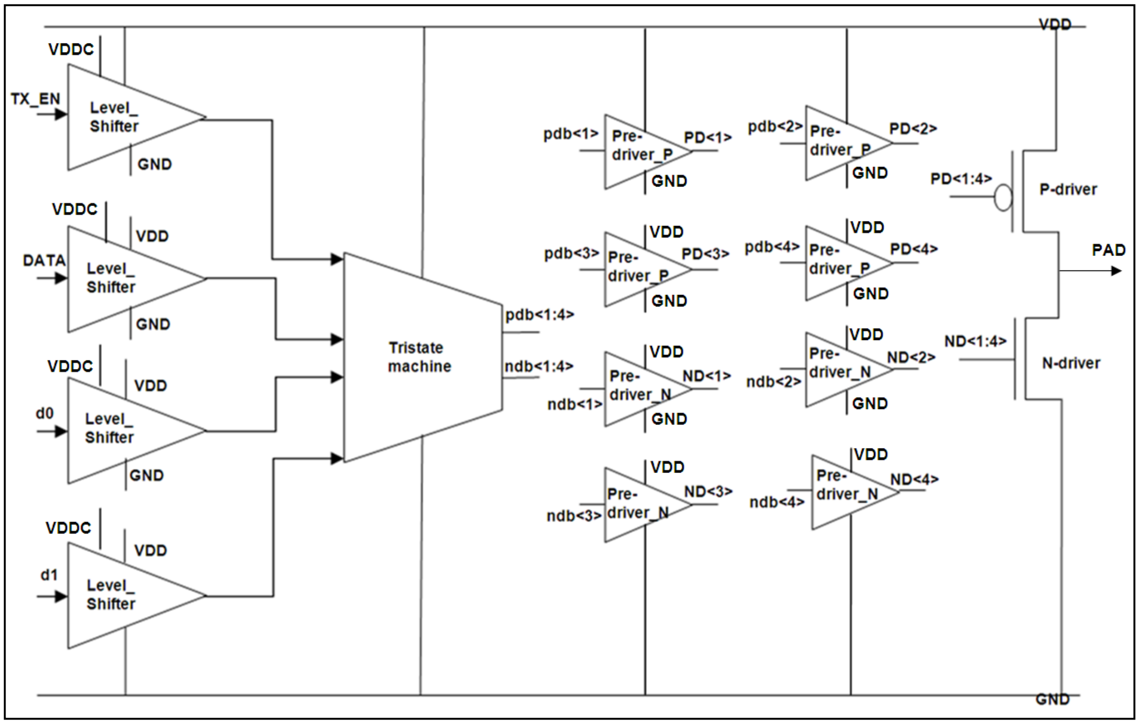


Figure 2-4 Top-level schematic of driver (VDD represents IO-level voltage)

2.3.1 Design of P-driver and N-driver:

According to the specifications given in Table 2-1, the driver should be able to drive at least 16 mA in the worst Process Voltage Temperature (PVT) corner (slow, 1.62 V and 125 °C). The width (size) of the P-driver is designed by connecting VOH at the PAD and varying the width until the driver is able to source 16 mA of current at the worst operating condition, starting from least possible width for the technology. Similar procedure is followed to design the N-driver by connecting VOL at the PAD.

The design of P-driver is explained arithmetically as below. Schematic setup for the simulation is shown in Figure 2-5. If RON represents the on resistance of the transistor,

$(VDDIO - VOH)/RON = 16mA$, the drive current required in the worst simulation corner.

From Table 2-1, for worst corner case, VDDIO = 1.62 V and VOH = 1.42 V.

$Ron = 0.2V/16mA$ = 12.5 Ω

But, $Ron = 1/\beta p(VDDIO - |VTp|)$ (1)

and $\beta p = \mu pCox(W/L)p$

In the equation [6], µp is the mobility of holes, COX is the capacitance of the gate oxide of the PMOS transistor, W and L refers to the width and length of the transistor respectively, VTp is the threshold voltage of the PMOS transistor. In Equation (1), W is the only variable since all other quantities are either fixed for a particular PDK or fixed for this particular design experiment. So, the width (size) of the P-driver can be calculated from this equation. For sub-micron technologies, second-order effects also play a vital role in circuit response. So, circuit simulation is the most dependable approach to accurately perform circuit design. Since the current work is done with 28 nm PDK, design is carried out by performing circuit simulation using the HSPICE simulator. Similar equations and similar design approach are valid for NMOS transistor as well.

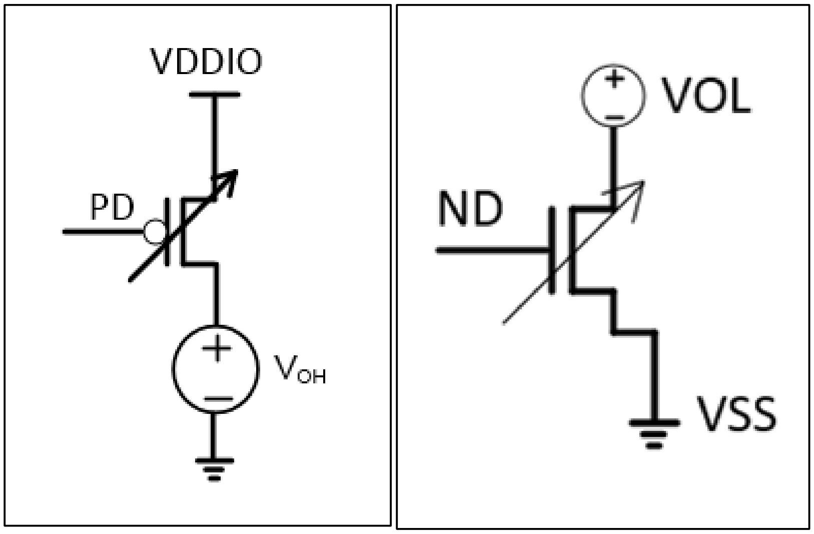
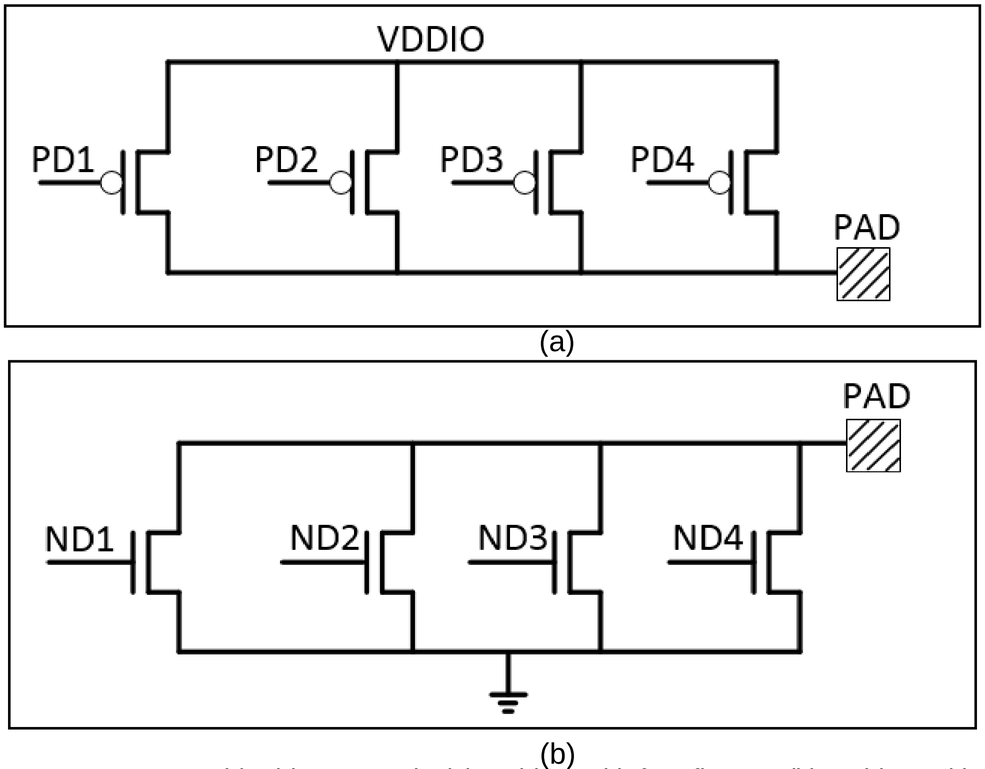


Figure 2-5 Simulation setup for P-driver and N-driver design

Total width (size) of both driver transistors are represented by four equal fingers as shown in Figure 2-6 so that the drive strength can be controlled. This arrangement provides drive strength options of 4 mA, 8 mA, 12 mA and 16 mA. Four dummy transistors are also added to both P-driver and N-driver schematics so that unexpected failure of transistors can be addressed. Figure 2-7 illustrates how the same load is charged by drivers with two different drive strength, 16 mA and 12 mA.



(a)

(b)

Figure 2-6 Programmable drive strength: (a) P-driver with four fingers, (b) N-driver with four fingers

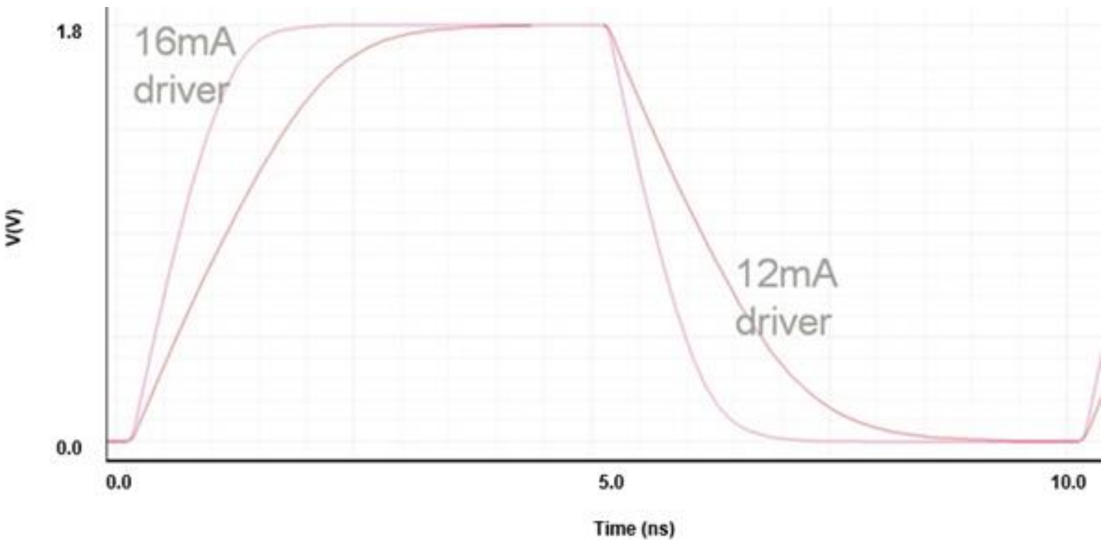


Figure 2-7 Drivers with different drive strength

As Figure 2-7 indicates, reduced drive strength may be sufficient if the load impedance is low.

2.3.2 Design of Pre-driver:

In order to drive the strong – geometrically wide – driver, a pre-driver is necessary. The pre-driver interfaces the small core logic circuit to the large capacitances – includes the gate capacitance and the parasitic capacitance of the metal route – of the driver. Pre-driver must be able to drive the large capacitance in the driver transistors. Sizing of pre-driver transistors is adjusted such that their rise time matches with the fall time for typical PVT conditions. The circuit arrangement for pre-driver is shown in Figure 2-8. The circuits used for both the P-driver and the N-driver are the same, but the sizes of transistors in each of them differ since the output load is different. Net names can be related to the net names shown in Figure 2-4.

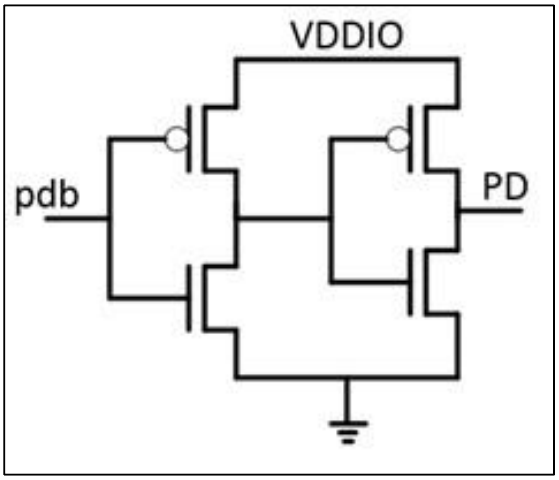


Figure 2-8 Pre-driver circuit for P-driver

2.3.3 Design of Tri-state Machine:

In the tri-state machine, the circuit which generates separate controls for the P-driver and the N-driver, incorporates a crowbar current control circuit. This is achieved by making sure that the already ON driver is turned OFF before turning ON the other driver. The logic function of the tri-state machine is basically a NAND and NOR operation with DATA_from_CORE and TX_EN as shown in Figure 2-9. The truth table is given in Table 2-4 and the circuit is shown in Figure 2-10. The signals PD_in and ND_in actually drive P-driver and N-driver respectively. Since the transistors in the tri-state machine circuit are too weak to drive the large transistors, PD_in and ND_in signals go through pre-driver circuit before driving the P-driver and the N-driver. The tri-state machine eventually helps to bring the PAD to high impedance state. The outputs – PD_in and ND_in signals – of the tri-state machine circuit for different combinations of DATA_from_CORE and TX_EN are shown in Figure 2-11. The crowbar current control waveforms are shown in Figure 2-12 and Figure 2-13.

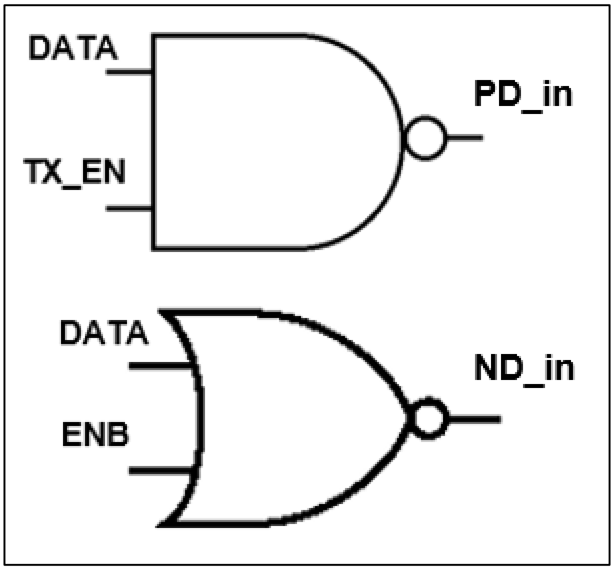


Figure 2-9 Logic representation of tri-state machine

Table 2-4 Truth table – Tri-state machine

DATA	TX_EN	ENB	PD_in	ND_in
X	0	1	1	0
X	0	1	1	0
0	1	0	1	1
1	1	0	0	0

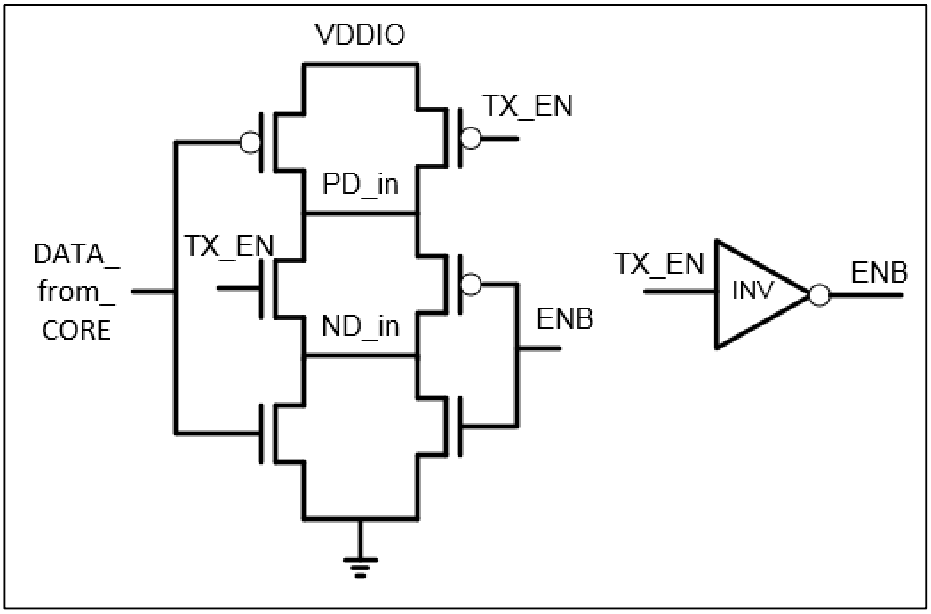


Figure 2-10 Circuit for tri-state machine

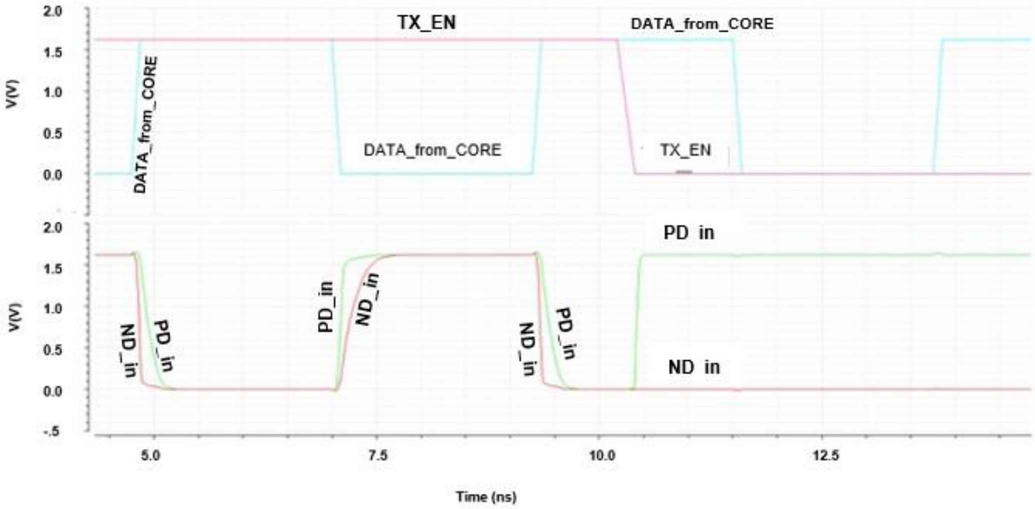


Figure 2-11 Tri-state machine waveforms

Slew rate control is implemented by having programmable resistance and capacitance in the circuit. If the circuit needs to be operated at a lower slew rate, then the programmable bit can be turned ON which will add additional R and C in the path so that the slew is worse.

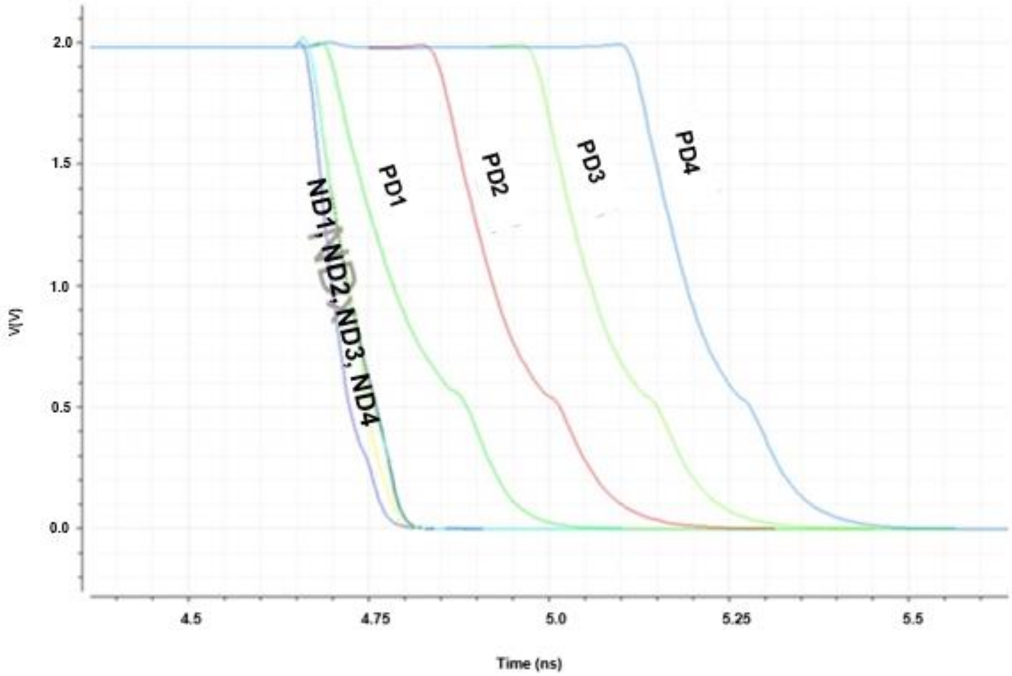


Figure 2-12 Crowbar current control for P-driver

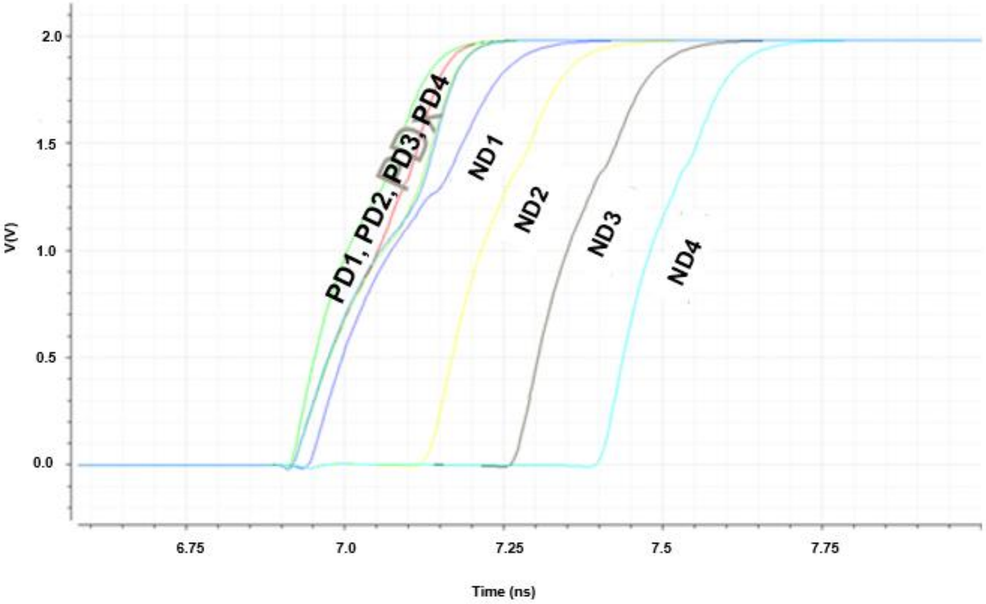


Figure 2-13 Crowbar current control for N-driver

2.3.4 Design of a Level-shifter:

Simple inverters do not work as level-shifters where the requirement is to up-convert the voltage. A standard level conversion CMOS circuit is used here in which the inverter circuit acts as a buffer for signals coming from the core (possibly a long route). The circuit is shown in Figure 2-14 and the voltage waveforms in Figure 2-15.

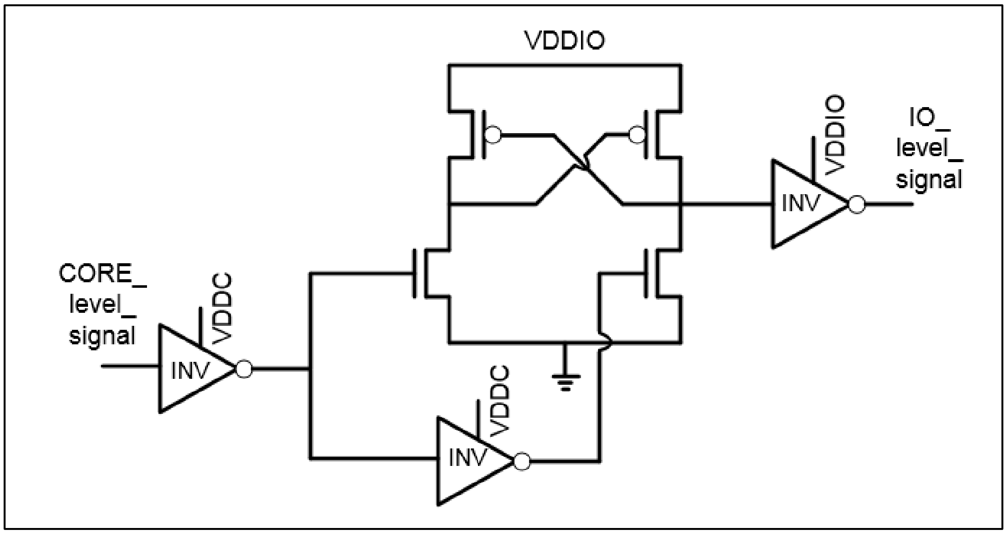


Figure 2-14 Level-shifter circuit

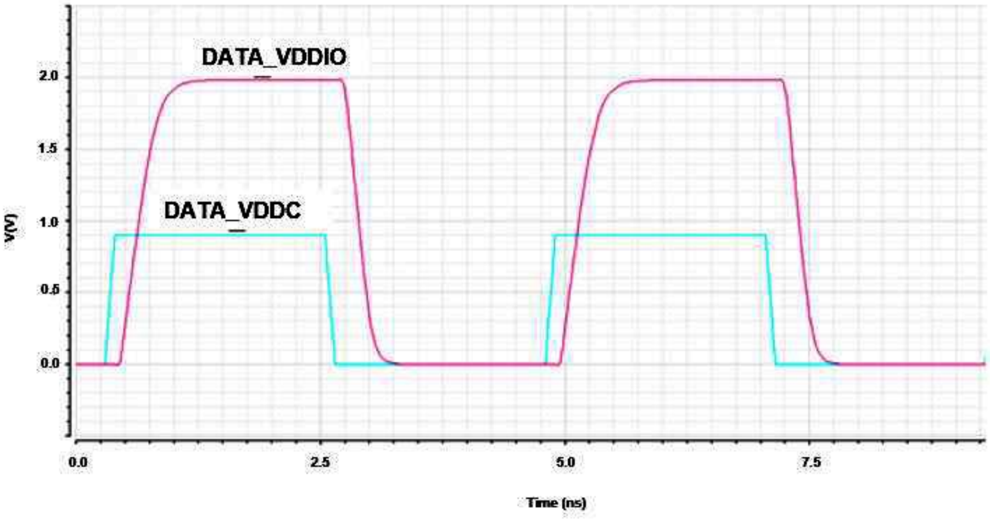


Figure 2-15 Level-shifter waveform

The circuit arrangement which is used to protect gate oxides from breakdown due to charge accumulation when gate of a small MOSFET is connected to a metal interconnect having a large area is called antenna diode. Such diodes are added to the long routes from the core to protect the CMOS devices. Decaps are added in layout wherever needed.

2.3.5 Design of Receiver:

The receiver block has a hysteresis control, a level-shifter and a core buffer. The hysteresis control circuit is designed such that 300 mV hysteresis is achieved. The receiver has an enable (RX_EN) signal which enables or disables the receiver operation. The receiver schematic is shown in Figure 2-16. The waveform for receiver circuit is shown in Figure 2-17.

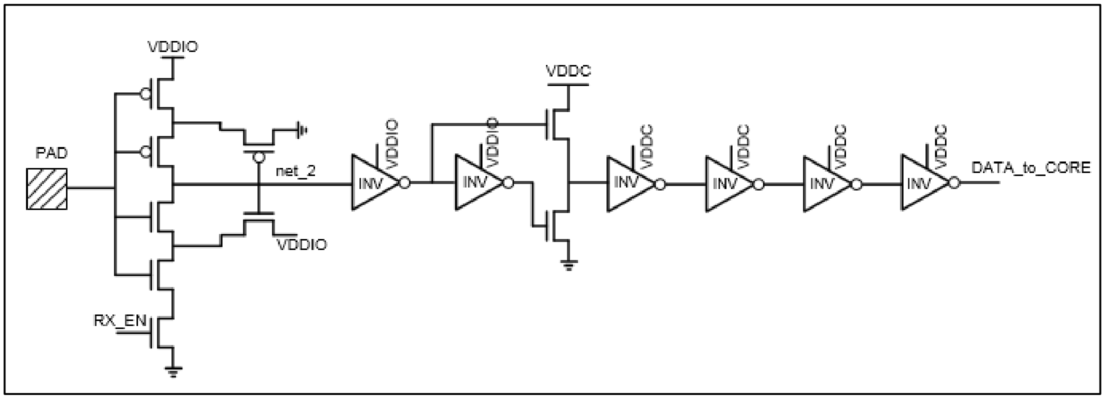


Figure 2-16 Schematic of receiver

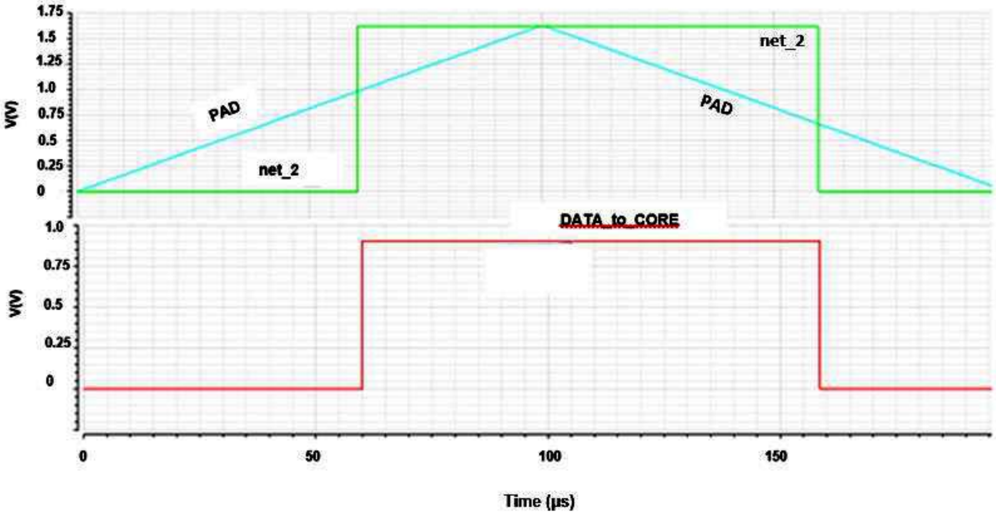


Figure 2-17 Receiver threshold

2.4 Layout

In the circuit layout it is extremely important to have good floor planning and power planning in place as well as a clean layout that passes DRC (Design Rule Check) and LVS (Layout Versus Schematic) checks. Floor plan for the GPIO in this work is illustrated in Figure 2-18.

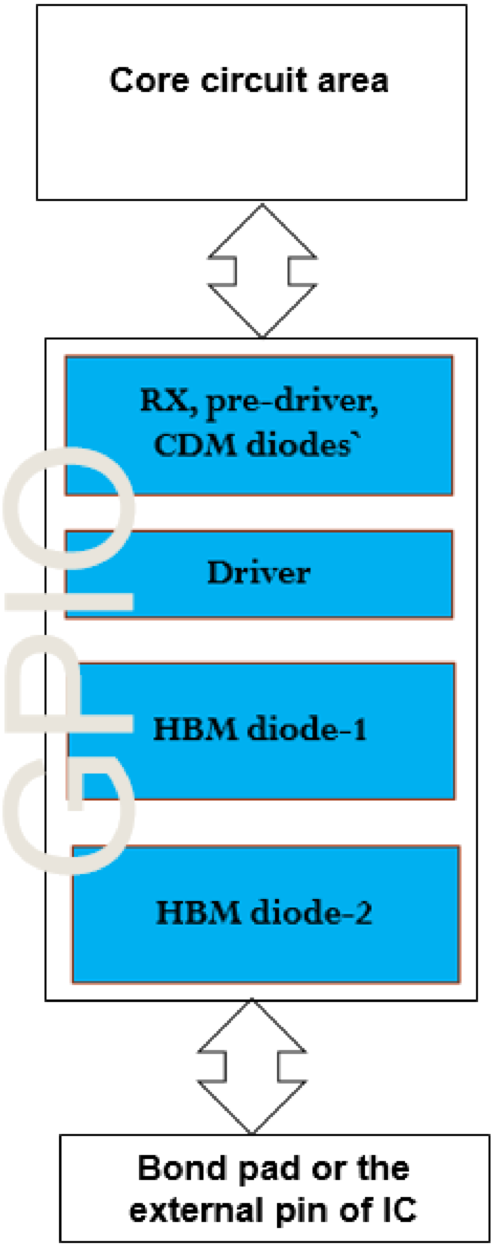

Core circuit area
RX, pre-driver, CDM diodes`
Driver
HBM diode-1
HBM diode-2
GPIO
Bond pad or the external pin of IC

Figure 2-18 Floor plan of the layout

Some of the things which are given emphasis while laying out the circuit are:

- Metallization is done up to Metal-3 for drivers so that it has the ability to handle the current even during a malfunction without breaking down.
- Drain routings for drivers extend as fingers and connect to the PAD which ensure that there is no break down due to inability to handle current.
- Substrate contacts and guard rings are provided so that chances of latch-up are reduced.
- ESD diodes are laid out as it is in the standard library.
- Decaps are added wherever possible, to help stabilize the power supply.
- Metal-1, Metal-2 and Metal-3 are used for signal routing.
- Metal-4, Metal-5, two layers of 6X metals (6 times wider than the minimum width metal) are used for power routing; care is taken to make the power plan as robust as possible by having wider metals and maximum number of vias.
- Pitch of the GPIO layout is 45 µm.
- The layout has no LVS or DRC errors.

2.5 Simulation

Parasitic extraction of the layout is performed and the resulting netlist (a textual representation of electronic circuit) is simulated to evaluate the circuit performance. The circuit is simulated for 270 different Process Voltage Temperature (PVT) conditions and circuit meets specifications for 99% of the conditions. The GPIO waveforms and the eye diagram of the signal at the PAD are shown in Figure 2-19 and Figure 2-20 respectively.

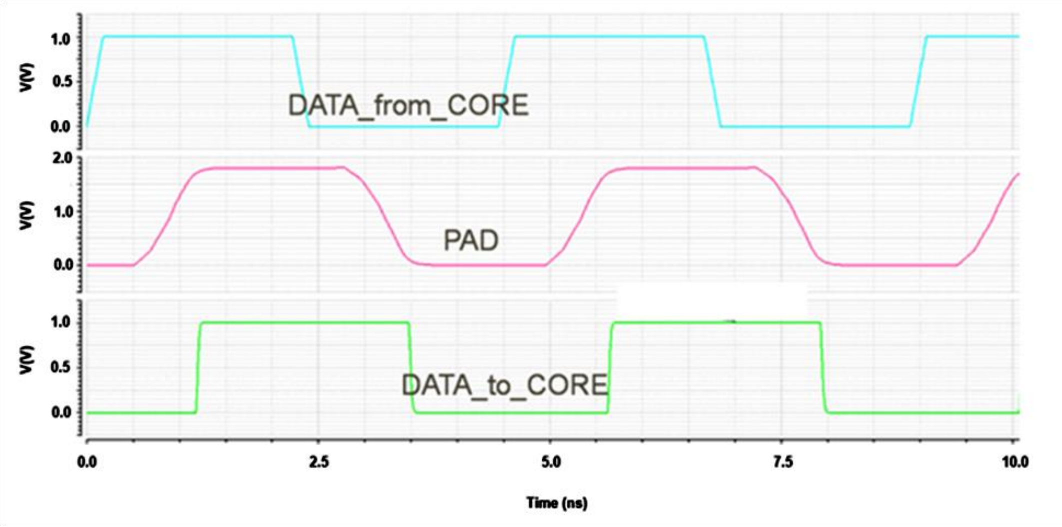


Figure 2-19 GPIO waveforms

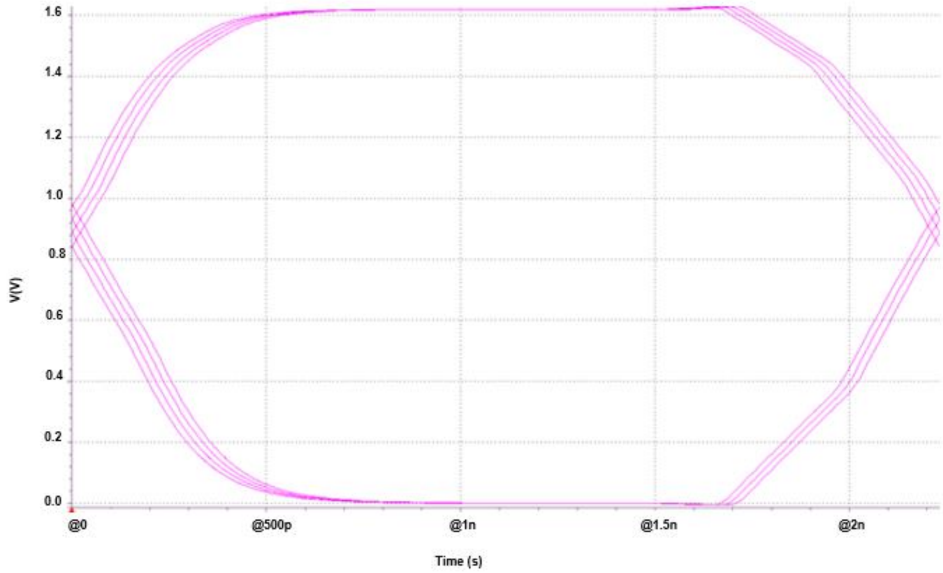


Figure 2-20 Eye diagram of the signal at the PAD

Chapter 3

Introduction to Electro Static Discharge (ESD) in ICs

ESD (Electrostatic discharge) failure is one of the critical reliability problems which has been present in the IC industry for a long time. ESD failure accounts for almost 35% of the total IC field failures and cost several billion dollars to the industry annually [7]. There has been external protection methods for ESD failures, but the electronics industry started adopting dedicated on-chip ESD protection methods some time ago. Active research on ESD fundamentals could answer many issues, but still there are grey areas to be addressed. Predictive ESD CAD design is one of the areas of recent interest. Since the development in the IC industry is fast paced, new ESD problems get introduced continuously [7]. The aim of the current work is to predict the optimum distance ("optimum distance" in this document represents the "maximum distance" at which the ESD clamp can be placed in an IO ring with reduced chances of ESD failure) for the placement of ESD clamps so that chances of an ESD failure is reduced. This helps save time, effort and money. An Electronic Design Automation (EDA) tool "IO Planner" is developed here so that the placement of clamps is done to a level where chances of IC failure due to ESD events is reduced. Any IC design engineer can use the "IO planner" tool as it does not demand deep ESD knowledge from the engineer.

An ESD event is basically a charge balancing act between two objects at different potentials. It can happen through direct contact or through induced electric field. ESD strikes from different sources is illustrated in Figure 3-1.

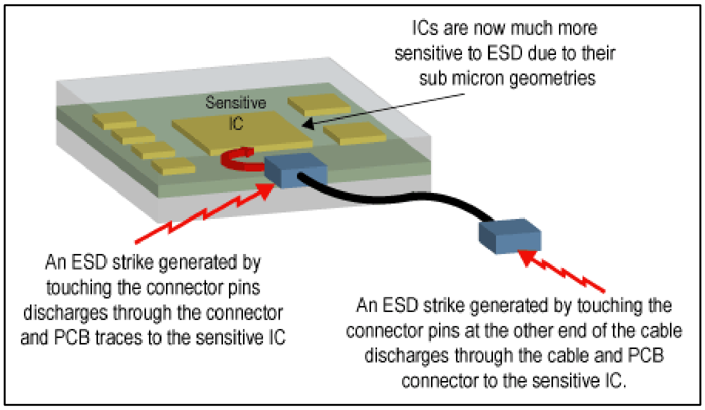

ICs are now much more sensitive to ESD due to their sub micron geometries
Sensitive IC
An ESD strike generated by touching the connector pins discharges through the connector and PCB traces to the sensitive IC
An ESD strike generated by touching the connector pins at the other end of the cable discharges through the cable and PCB connector to the sensitive IC.

Figure 3-1 ESD strikes from different sources. Source: http://www.hdcabling.co.za/ .

High voltage or current stress occurring for a very short interval of time is the characteristic of an ESD event. Thus the impact of the event is so high that it can damage the IC. The different damages caused by ESD failures are shown in Figure 3-2. Study of ESD is a field where electrical, thermal and mechanical engineering join hands. The solution to protect ICs from ESD events is to discharge the high current via a low impedance shunting path or clamp the PAD voltage to a sufficiently low level or ground.

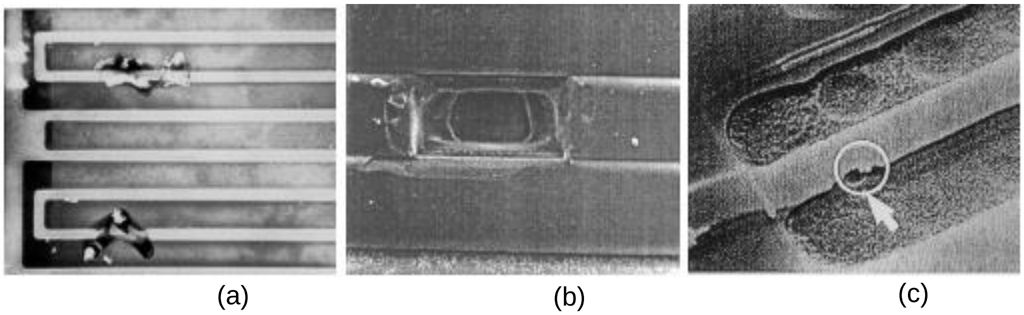


Figure 3-2 Damages caused by ESD failures: (a) junction breakdown, (b) metal/via damage, (c) gate oxide damage [8], [9]

3.1 ESD Test Models

Different test models, categorized by their origin, are used to simulate different ESD events upon which ESD protection circuits are tested and qualified. The models are the Human Body Model (HBM), the Machine Model (MM) and the Charged Device Model (CDM).

3.1.1 Human Body Model (HBM)

The HBM represents the ESD event which occurs when a charged human body or a charged material touches the electronic circuit. Charge transfer happens through physical contact. The equivalent model is shown in Figure 3-3. In the HBM standard [10], the circuit component used to simulate the charged human body is a 100 pF capacitor and the resistance of the discharging path is 1500 Ohm; it electrically looks like a current source if the Design Under Test (DUT) provides a current path of low resistivity. LHBM (~ 0.75uH) is the effective inductance of the discharge path in a real tester.

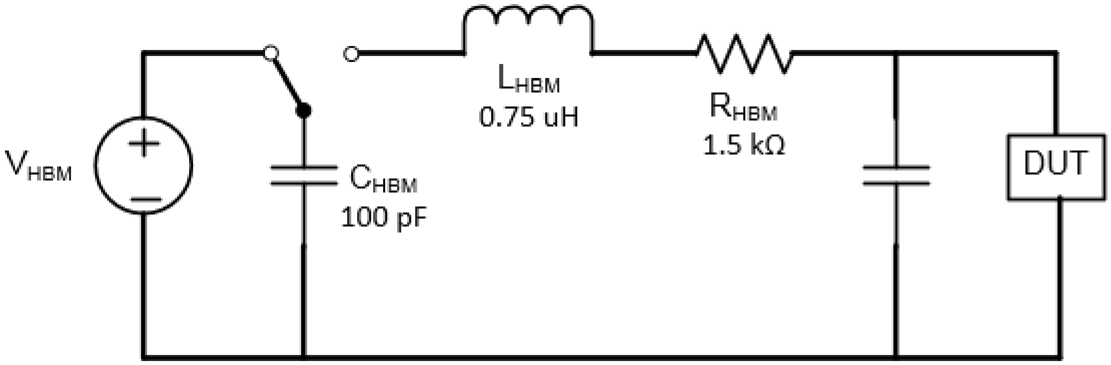


Figure 3-3 Circuit model to represent HBM

The rise time of the HBM pulse can be 5-10 ns (≈ 2 x L/R) and the decay time is around 150 ns (~= RC). HBM has the longest pulse among the three models. When a charged body comes in contact with the circuit, charge from the body gets transferred to the device under test (DUT). The HBM event leads to thermal destruction of the IC. A typical HBM waveform is shown in Figure 3-4.

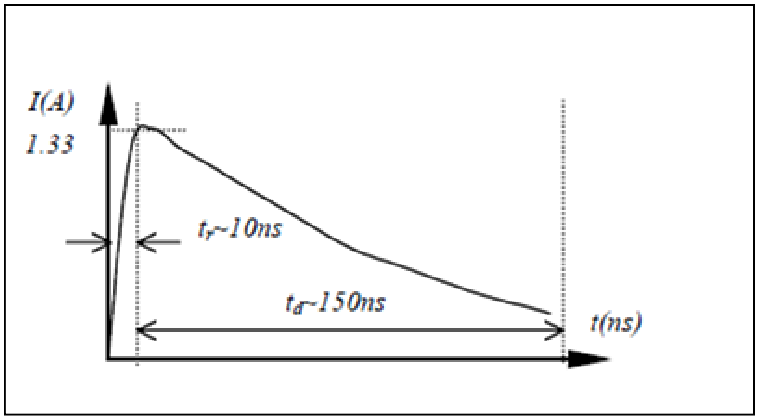


Figure 3-4 Typical HBM waveform [7]

3.1.2 Machine Model (MM)

The machine model represents the ESD event which occurs where charged machinery discharges while touching the IC pins during testing, packaging or any handling. The MM represents a worst-case HBM where the peak current is higher than that in an HBM and the rise time is shorter. Similar to an HBM event, a capacitor (200 pF) which represents a conductive object such as a metallic handler is charged up to a high voltage and then discharged through the pins of an IC. In this model, it is assumed that an arc discharge occurs between the charged source and the DUT [11]. The MM circuit model is similar to HBM circuit model with different values for circuit components and is shown in Figure 3-5.

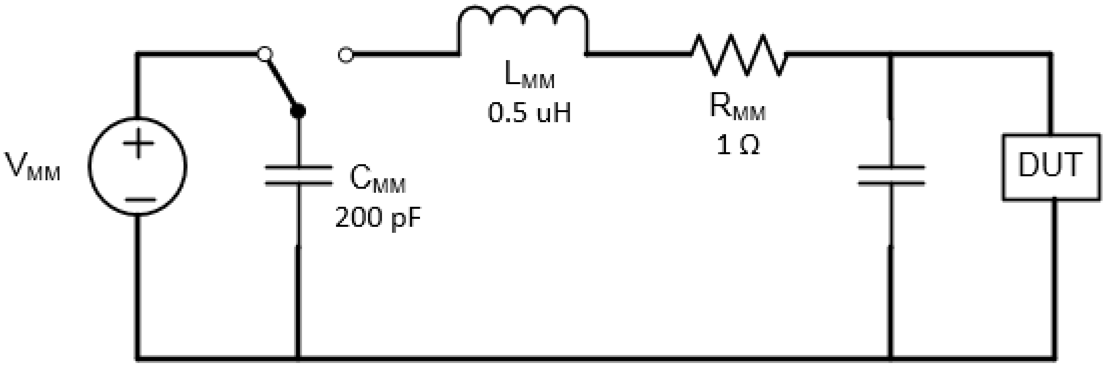


Figure 3-5 Circuit model to represent MM

An arc discharge fundamentally has a resistance of 10-20 Ω which is much lower than the RHBM (~ 1500 Ω). Therefore the MM response is more rapid than the HBM event and has a form of bidirectional damped oscillation. The discharge process and failure signatures are generally the same as that of an HBM event. A typical MM waveform is shown in Figure 3-6.

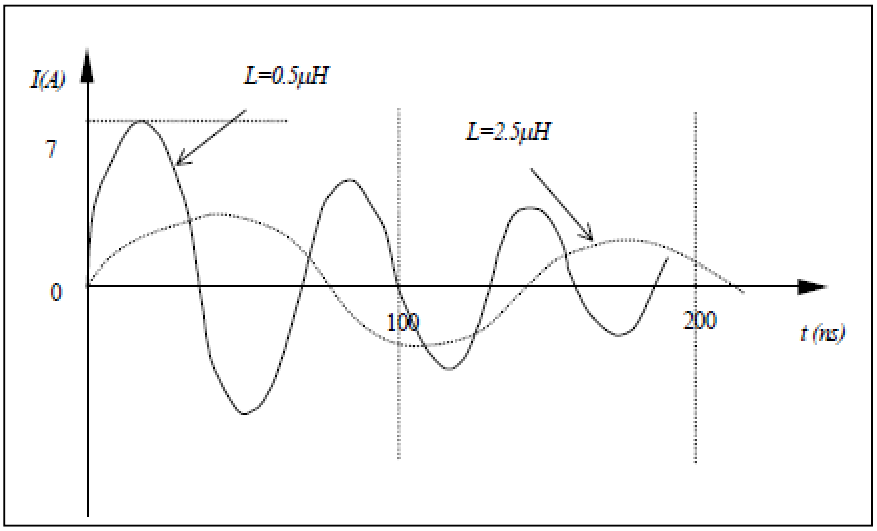


Figure 3-6 Typical MM waveform [7]

3.1.3 Charged Device Model (CDM)

The charged device model simulates the event where the un-grounded electronic parts accumulate charge during manufacturing or assembly and then discharge to ground. Unlike the HBM and MM event, CDM event involves a single pin on the module. The circuit model used to represent a CDM event is shown in Figure 3-7. The charging process can be initiated by direct charging or field induced charging. The discharge process can be initiated as an electric contact is made between the charged device and a discharging means such as automated handlers. As shown in Figure 3-8, the CDM event occurs in an extremely short interval (~ 1 ns) but generates very high current (~7 A). The rise time of the event is of the order of 250 ps. The CDM event can lead to dielectric failure.

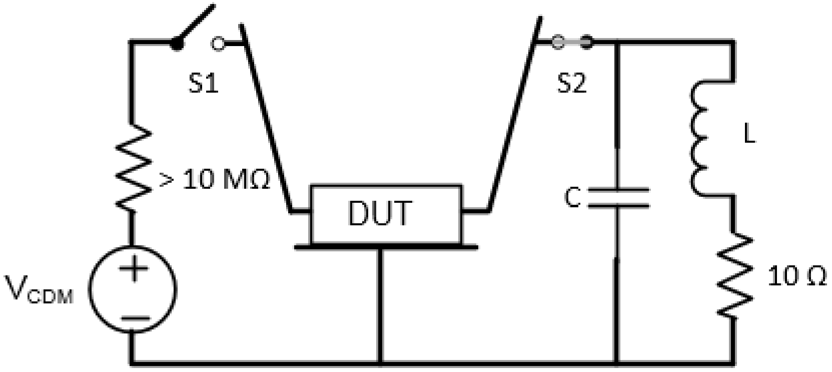


Figure 3-7 Circuit model to represent CDM

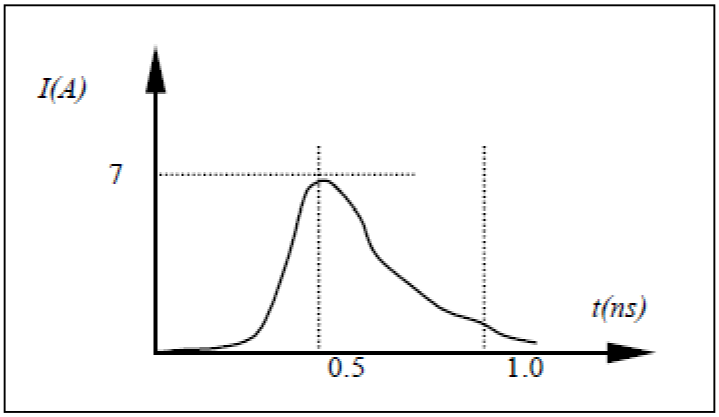


Figure 3-8 Typical CDM waveform [7]

A comparison of typical HBM, MM and CDM waveforms are shown in Figure 3-9.

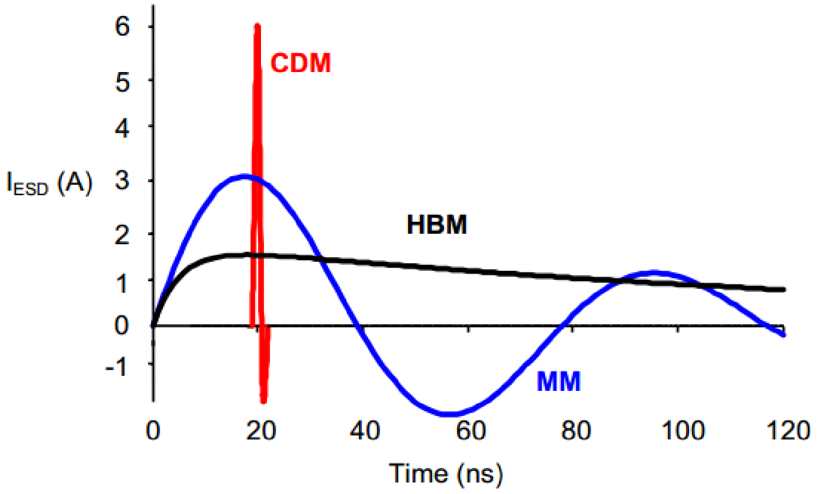


Figure 3-9 Comparison of HBM, CDM and MM waveforms [12]

3.2 ESD Protection Schemes

Since ESD failures are too costly to ignore, devising protection methods is critical. There can be two different approaches. One is to reduce the amount of ESD induced charges and redistribute them through proper handling of devices. Another approach is to implement on-chip ESD protection circuits. Most of the on-chip ESD solutions rely on shunting charge from an IO pin to a power supply. ESD protection schemes are placed at each IO and supply pins. The two popular types of devices used for ESD protection are discussed below.

3.2.1 Turn-on Type Device

A turn-on device like diode turns-on after reaching a particular trigger-voltage. Once the device is turned-on, it offers a low impedance path for the ESD current to flow. The Current-Voltage (IV) characteristic for such a device is shown in Figure 3-10.

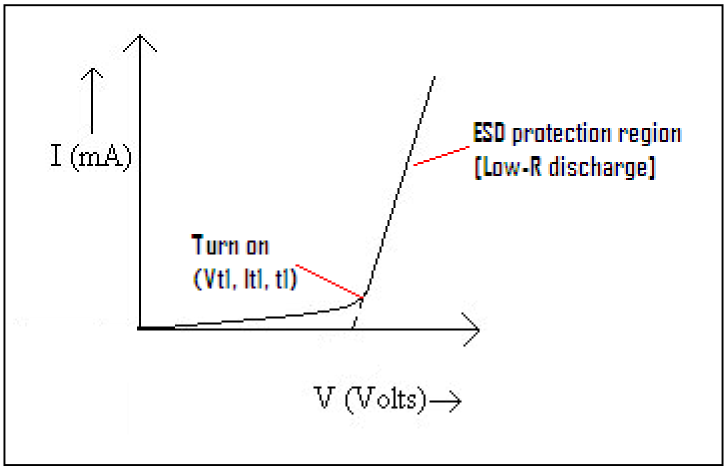


Figure 3-10 I-V characteristics for a turn-on type device

In Figure 3-10, t1 represents the turn-on time and Vt1 is the trigger voltage.

Diodes offer a simple and effective turn-on type ESD protection. They can be used in either the forward-biased or reverse-biased configuration since diodes have the I-V characteristics as shown in Figure 3-11.

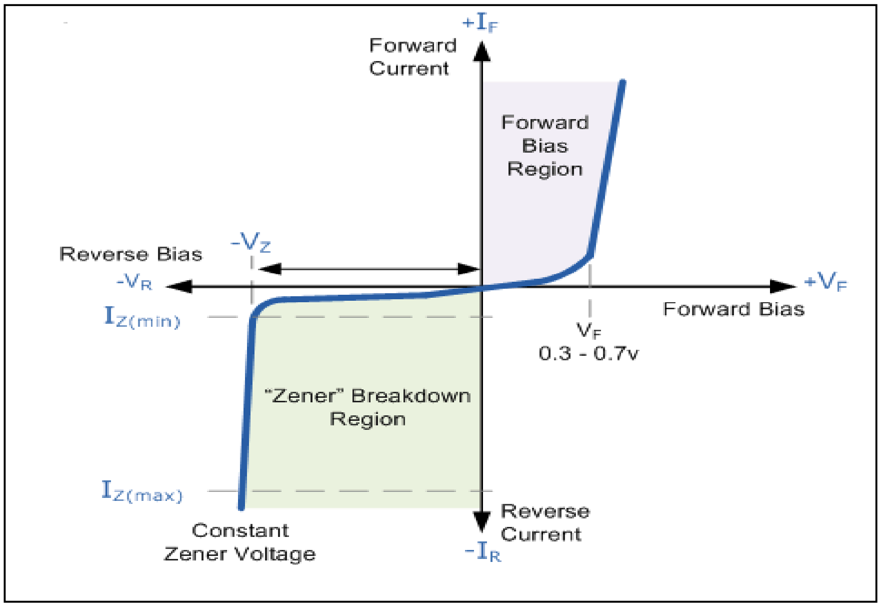


Figure 3-11 I-V characteristics of a diode

They offer a low resistance path beyond VF or -VZ, which are marked in Figure 3-11. The advantage of this method is that it can be simulated using SPICE. One limitation is the fixed diode turn-on voltage which can reduce the application of the diode for this.

3.2.2 Snapback Type Device

Snapback type protection schemes are capable of handling higher currents. Typical IV characteristics are given in Figure 3-12.

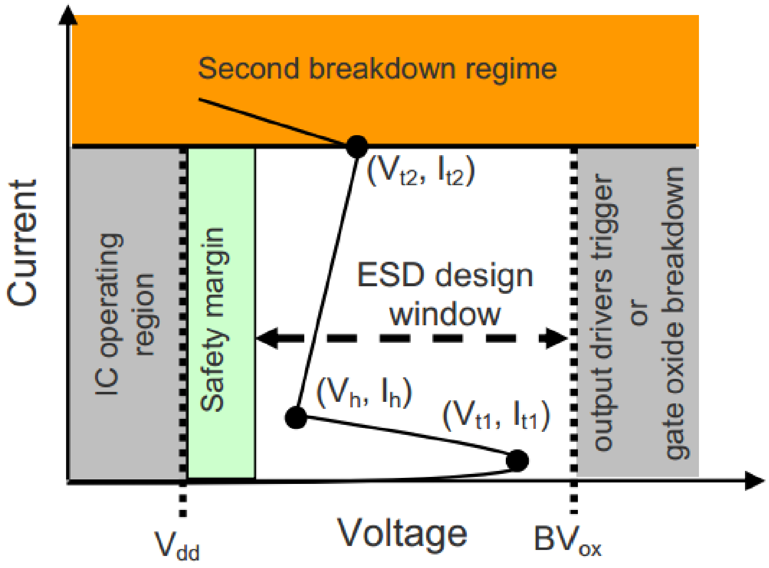


Figure 3-12 I-V characteristics for a snapback type device

A good design defines the critical parameters like the triggering point (Vt1,It1), snapback holding voltage and current corresponding to the voltage, Vh (Vh,Ih) and thermal breakdown voltage and current corresponding to the voltage, Vt2 (Vt2,It2). Grounded-gate NMOS (ggNMOS), gate-coupled NMOS (gcNMOS), SCR (Silicon controlled rectifier) are ESD protection devices which work based on snapback mechanism.

3.2.3 ESD Protection Circuit

There are different on-chip protection methods used for ESD protection. The rail-based ESD protection circuit is one among them. In general, this scheme has a primary network, secondary network and an RC-triggered power clamp. A typical arrangement is shown in Figure 3-13 indicating a current path for a positive ESD excitation.

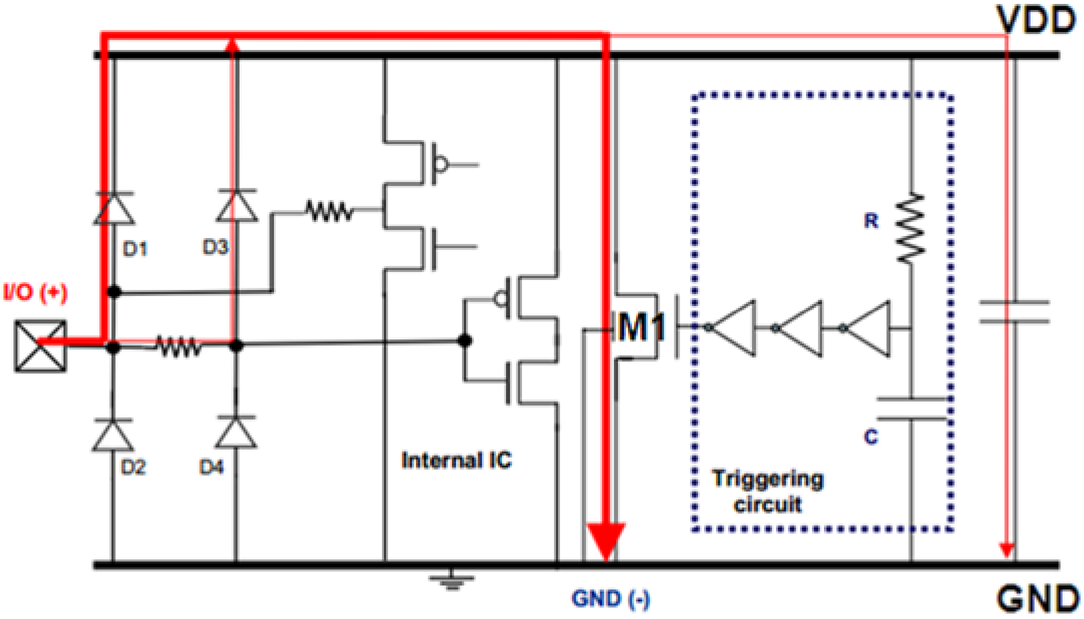


Figure 3-13 A typical ESD protection circuit [12]

Here, diodes D1, D2 form the primary network and D3, D4 form the secondary network. M1 represents the clamping device which is triggered by the RC arrangement and three series inverters.

When the positive ESD pulse is induced on the PAD, there is no direct path from the PAD to GND; diodes D1 and D3 are triggered on and the ESD current flows through them to the power supply line and to GND through the triggered power clamp. Design of each of the involved devices is done such that it is able to handle the ESD current. For a negative ESD pulse, current enters the supply line, flows through the power clamp and through the diodes D2, D4 to the GND line.

Placement of the power supply clamps determines the effectiveness of the IO protection in a rail-based ESD protection arrangement. Therefore, it is advantageous if the optimum distance to place power clamps can be estimated so that the probability of failure in silicon is reduced. The work presented here analyzes the ESD network and suggests the optimum distance to place power clamps.

3.3 Power Clamp

Power clamps are present inside each power supply IO. They are used to shunt current to ground from power supply lines when an ESD event occurs or any event which can potentially damage the electronic components. At the least, a chip will have three such pins namely, VDDIO (IO-level power supply), VDDC (core-level power supply) and VGND (ground for both supplies). The RC-triggered power clamp is a simple and efficient implementation to achieve the same. One such typical RC-triggered power clamp is given in Figure 3-14.

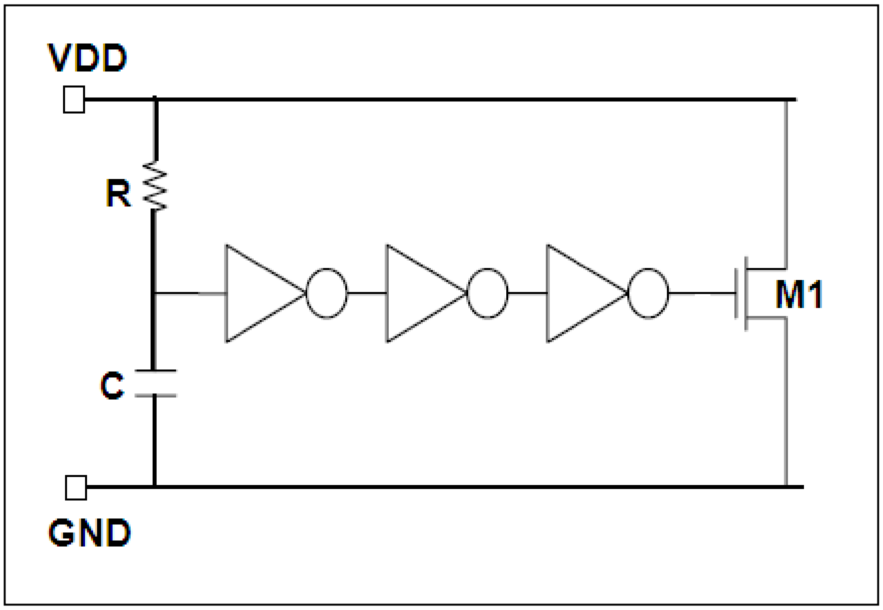


Figure 3-14 RC-triggered power clamp

Design of the RC network and the 3-stage inverter circuit is done according to the trigger requirements for the clamp. The width of M1 (size of the transistor) is based on the maximum value of current it will have to shunt.

Chapter 4

ESD Network Analysis

An ESD network analysis is done here in order to suggest the optimum distance for power clamp placement. The aim of the approach is to find out how much margin is available for the voltage drop at a power bus. According to the available margin, the optimum distance at which a power clamp can be placed is suggested. Components of an ESD network are shown in Figure 4-1. Only the parasitic resistance is considered for this analysis since parasitic inductance is very small in GPIOs and the parasitic capacitance value which may exist would actually take up some current and it can only reduce the chances of ESD related failure.

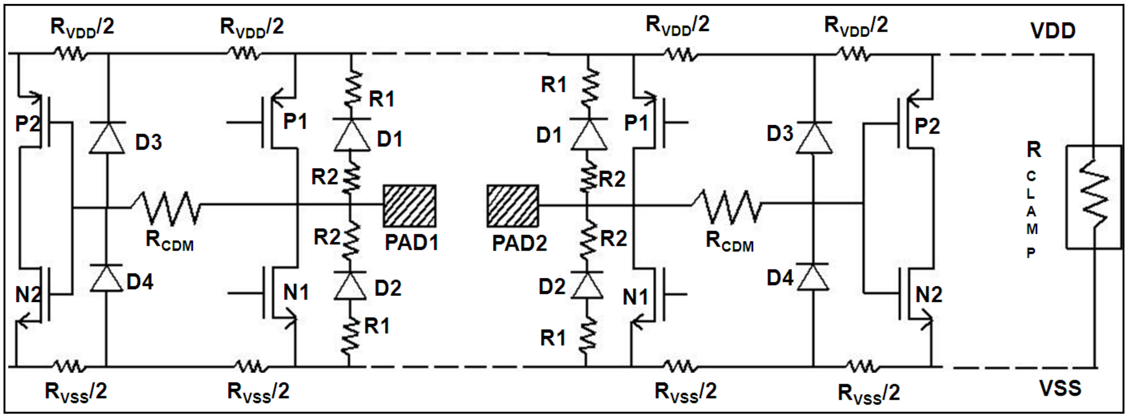


Figure 4-1 ESD network

The circuit components shown in the network are described below.

D1, D2 → Primary ESD protection network. D1 is referred to the HBM up-diode and D2 refers to the HBM down-diode in the following sessions of the report.

D3, D4 → Secondary ESD protection network. D3 is referred to the CDM up-diode and D4 refers to the CDM down-diode in the following sessions of the report.

R_{CDM} → CDM resistance

R_{CLAMP} → Clamp resistance

R_{VDD} → VDD bus resistance

R_{VSS} → VSS bus resistance

R1 → Diode to supply parasitic resistance

R2 → Diode to PAD parasitic resistance

P1 → PMOS driver

N1 → NMOS driver

P2, N2 → Receiver transistors

Two failure mechanisms for transistors are considered here. They are junction breakdown (source-drain junction) and gate-oxide breakdown during both an HBM event and a CDM event, for both PMOS and NMOS transistors associated with it. Hence, there are three variables available and there can be up to eight unique cases to be considered to ensure that the IC does not fail due to an ESD event. When there is an HBM event, current flow is from PAD to PAD, whereas for a CDM event, it is from PAD to VSS. Kirchhoff's voltage law is applied for the loop related to each case and the maximum possible value for R_{VDD} and R_{VSS} is calculated. By knowing the resistance per µm for power buses, optimum distance to place power clamps is calculated and suggested.

Figure 4-2 shows the current flow direction for an HBM event for which a junction breakdown parameter for NMOS driver (N1 of IO-1) is evaluated. The HBM pulse originates from PAD1 and flows to PAD2 through R2, D1, R1, R_{VDD}, R_{CLAMP}, R_{VSS} and D2.

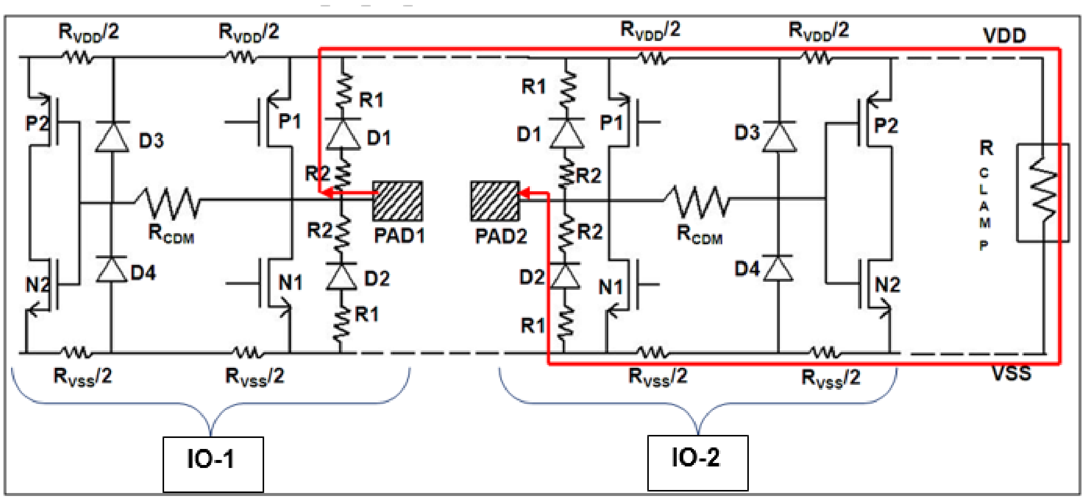


Figure 4-2 Current flow through ESD network during HBM event

Figure 4-3 indicates the current flow direction for a CDM event for which the gate-oxide breakdown parameter for the PMOS transistor (P2 of IO-1) is evaluated. The CDM pulse originates from VDD and a major portion flows to PAD1 through R_{CLAMP}, R_{VSS}, R1, D2, R2. A very small amount of current (~0.1%) takes the path through D4, R_{CDM} from R_{VSS}.

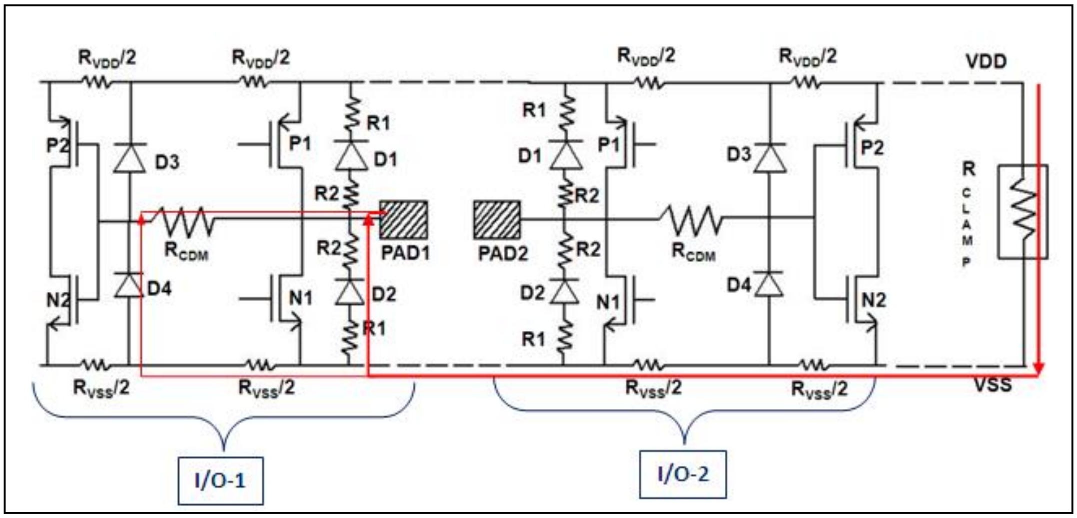


Figure 4-3 Current flow through ESD network during CDM event

Chapter 5

The “IO Planner” Tool

Based on the ESD network analysis explained in chapter 4, an EDA tool has been developed that finds the optimum position where a power clamp should be placed to reduce the risk of ESD failure. This tool works only for the diode based protection scheme. The tool considers two categories of implementation where one IO is supplied by one clamp and where one IO is supplied by two clamps. A pictorial representation of these cases is shown in Figure 5-1 and Figure 5-2.

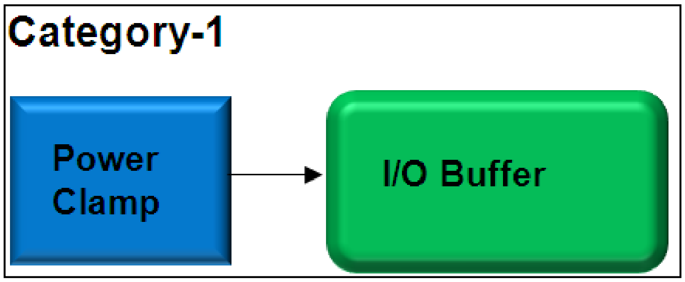


Figure 5-1 Category-1 arrangement

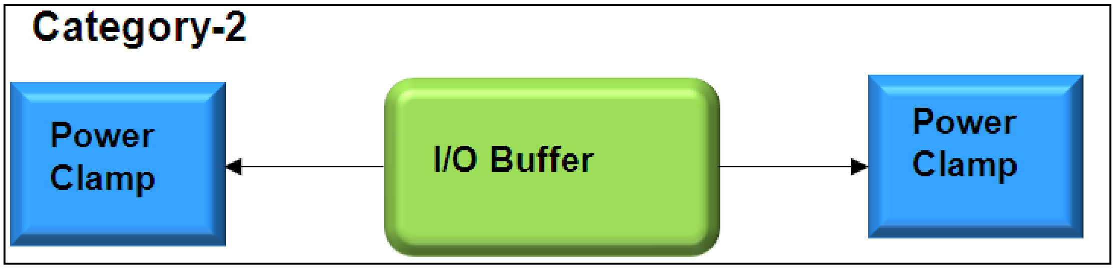
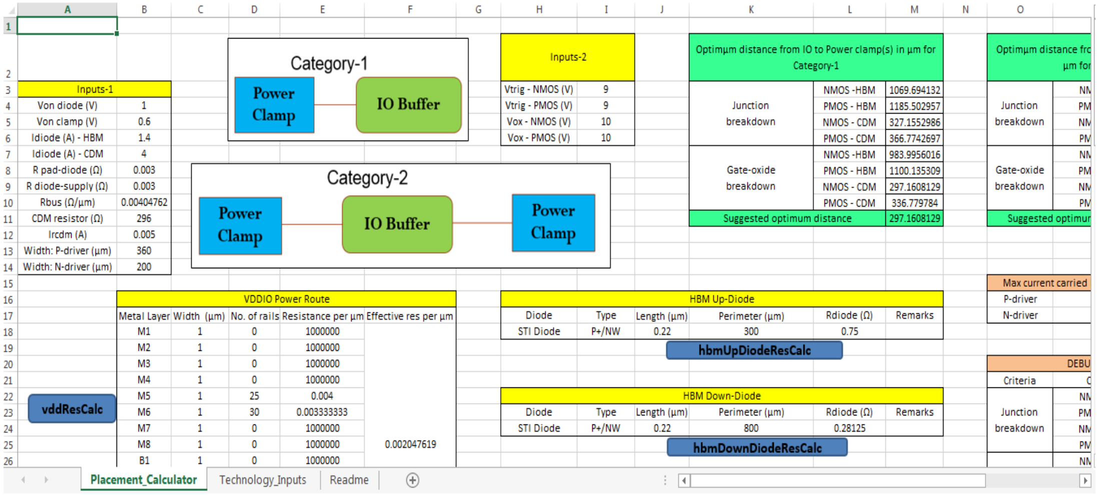


Figure 5-2 Category-2 arrangement

The program does ESD network analysis for 8 different cases shown in Table 5-1 for both category-1 and category-2 type of arrangements and suggest one optimum distance each for the two categories. The user interface for the program is shown in Figure 5-3; it takes user inputs and processes it to suggest the placement of power clamps.

Table 5-1 List of cases being analyzed

Type of failure	Device – event	Remarks
Junction breakdown	NMOS -HBM	This case analyze the junction breakdown of NMOS driver transistor during an HBM event
	PMOS - HBM	This case analyze the junction breakdown of PMOS driver transistor during an HBM event
	NMOS - CDM	This case analyze the junction breakdown of NMOS driver transistor during an CDM event
	PMOS - CDM	This case analyze the junction breakdown of PMOS driver transistor during an CDM event
Gate-oxide breakdown	NMOS -HBM	This case analyze the gate-oxide breakdown of NMOS receiver transistor during an HBM event
	PMOS - HBM	This case analyze the gate-oxide breakdown of PMOS receiver transistor during an HBM event
	NMOS - CDM	This case analyze the gate-oxide breakdown of NMOS receiver transistor during an CDM event
	PMOS - CDM	This case analyze the gate-oxide breakdown of PMOS receiver transistor during an CDM event



Inputs-1	
Von diode (V)	1
Von clamp (V)	0.6
Idiode (A) - HBM	1.4
Idiode (A) - CDM	4
R pad-diode (Ω)	0.003
R diode-supply (Ω)	0.003
Rbus (Ω/µm)	0.00404762
CDM resistor (Ω)	296
Ircdm (A)	0.005
Width: P-driver (µm)	360
Width: N-driver (µm)	200

Inputs-2	
Vtrig - NMOS (V)	9
Vtrig - PMOS (V)	9
Vox - NMOS (V)	10
Vox - PMOS (V)	10

Optimµm distance from IO to Power clamp(s) in µm for Category-1		
Junction breakdown	NMOS -HBM	1069.694132
	PMOS - HBM	1185.502957
	NMOS - CDM	327.1552986
	PMOS - CDM	366.7742697
Gate-oxide breakdown	NMOS -HBM	983.9956016
	PMOS - HBM	1100.135309
	NMOS - CDM	297.1608129
	PMOS - CDM	336.779784
Suggested optimum distance		297.1608129

Optimµm distance fro µm for	
Junction breakdown	NM
	PM
	NM
	PM
Gate-oxide breakdown	NM
	PM
	NM
	PM
Suggested optimur	

VDDIO Power Route				
Metal Layer	Width (µm)	No. of rails	Resistance per µm	Effective res per µm
M1	1	0	1000000	
M2	1	0	1000000	
M3	1	0	1000000	
M4	1	0	1000000	
M5	1	25	0.004	
M6	1	30	0.003333333	
M7	1	0	1000000	
M8	1	0	1000000	0.002047619
B1	1	0	1000000	

HBM Up-Diode					
Diode	Type	Length (µm)	Perimeter (µm)	Rdiode (Ω)	Remarks
STI Diode	P+/NW	0.22	300	0.75	

HBM Down-Diode					
Diode	Type	Length (µm)	Perimeter (µm)	Rdiode (Ω)	Remarks
STI Diode	P+/NW	0.22	800	0.28125	

Max current carried	
P-driver	
N-driver	

DEBU	
Criteria	C
Junction breakdown	NM
	PM
	NM
	PM
	NM

Figure 5-3 “IO planner” interface

This tool is developed using Microsoft-Excel (MS-Excel). The user-interface is MS-Excel worksheets and the programming is done using Excel-macro (Visual Basic).

The excel document has three sheets, namely “Placement_Calculator”, “Technology_Inputs”, and “Readme”. The inputs corresponding to each test case (ie. IO ring) can be provided in the Placement_Calculator sheet. Results of the ESD network analysis and the optimum placement distance is shown in the same sheet. The Technology_Inputs sheet contains the values corresponding to metal routes, diodes and clamps which are Process Development Kit (PDK) specific as shown in Figure 5-4. The readme sheet contains version history and other basic information about the tool as shown in Figure 5-5.

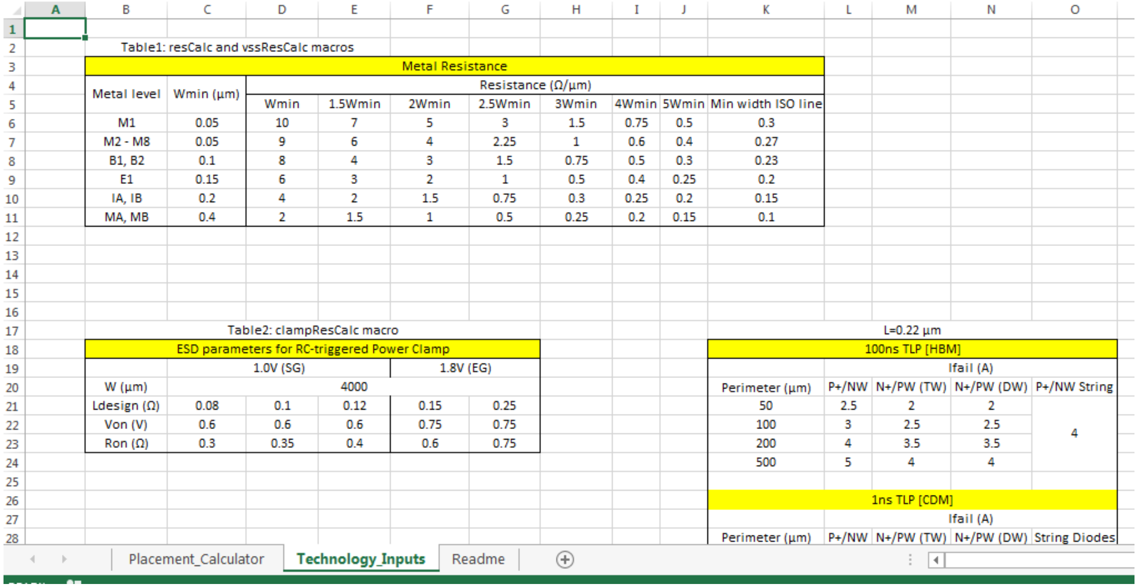


Table1: resCalc and vssResCalc macros

Metal Resistance									
Metal level	Wmin (µm)	Resistance (Ω/µm)							
		Wmin	1.5Wmin	2Wmin	2.5Wmin	3Wmin	4Wmin	5Wmin	Min width ISO line
M1	0.05	10	7	5	3	1.5	0.75	0.5	0.3
M2 - M8	0.05	9	6	4	2.25	1	0.6	0.4	0.27
B1, B2	0.1	8	4	3	1.5	0.75	0.5	0.3	0.23
E1	0.15	6	3	2	1	0.5	0.4	0.25	0.2
IA, IB	0.2	4	2	1.5	0.75	0.3	0.25	0.2	0.15
MA, MB	0.4	2	1.5	1	0.5	0.25	0.2	0.15	0.1

Table2: clampResCalc macro

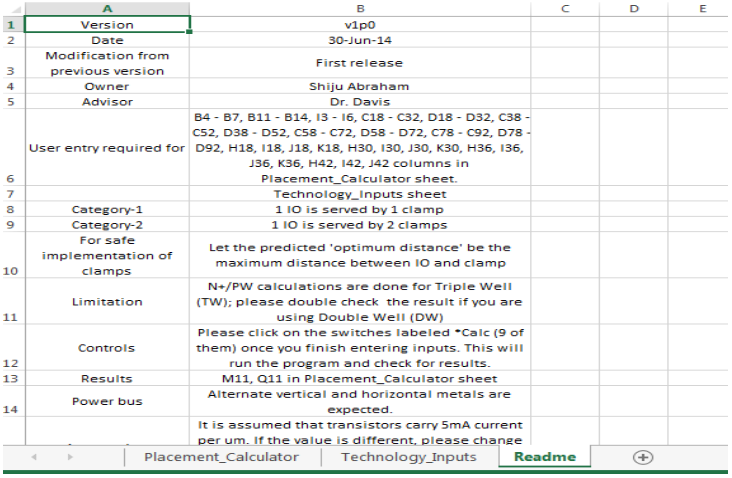
ESD parameters for RC-triggered Power Clamp					
	1.0V (SG)			1.8V (EG)	
W (µm)	4000				
Ldesign (Ω)	0.08	0.1	0.12	0.15	0.25
Von (V)	0.6	0.6	0.6	0.75	0.75
Ron (Ω)	0.3	0.35	0.4	0.6	0.75

L=0.22 µm

100ns TLP [HBM]				
	Ifail (A)			
Perimeter (µm)	P+/NW	N+/PW (TW)	N+/PW (DW)	P+/NW String
50	2.5	2	2	4
100	3	2.5	2.5	
200	4	3.5	3.5	
500	5	4	4	

1ns TLP [CDM]				
	Ifail (A)			
Perimeter (µm)	P+/NW	N+/PW (TW)	N+/PW (DW)	String Diodes

Figure 5-4 "Technology_Inputs" sheet in the IO Planner



Version	v1p0
Date	30-Jun-14
Modification from previous version	First release
Owner	Shiju Abraham
Advisor	Dr. Davis
User entry required for	B4 - B7, B11 - B14, I3 - I6, C18 - C32, D18 - D32, C38 - C52, D38 - D52, C58 - C72, D58 - D72, C78 - C92, D78 - D92, H18, I18, J18, K18, H30, I30, J30, K30, H36, I36, J36, K36, H42, I42, J42 columns in Placement_Calculator sheet.
	Technology_Inputs sheet
Category-1	1 IO is served by 1 clamp
Category-2	1 IO is served by 2 clamps
For safe implementation of clamps	Let the predicted 'optimum distance' be the maximum distance between IO and clamp
Limitation	N+/PW calculations are done for Triple Well (TW); please double check the result if you are using Double Well (DW)
Controls	Please click on the switches labeled *Calc (9 of them) once you finish entering inputs. This will run the program and check for results.
Results	M11, Q11 in Placement_Calculator sheet
Power bus	Alternate vertical and horizontal metals are expected.
	It is assumed that transistors carry 5mA current per um. If the value is different, please change

Figure 5-5 "Readme" sheet in the IO Planner

The voltage drop across the diodes [Von diode], the Voltage drop across the power clamp device [Von clamp], the maximum current expected for the HBM event [Idiode – HBM], the maximum current anticipated for the CDM event [Idiode – CDM], the value of the CDM resistor [CDM resistor], current through the CDM resistor [Ircdm], width of the P-driver and the width of the N-driver are the primary inputs a user has to enter in the Placement_Calculator sheet. Junction breakdown voltages and gate oxide breakdown voltages for both NMOS and PMOS transistors are also required [Vtrig – NMOS, Vtrig – PMOS, Vox – NMOS, Vox – PMOS]. The ESD network analysis carried out in the tool are split into different tasks and for the ease of coding and they are explained in sections 5.1 – 5.12.

In order to use the “IO planner”, please make sure that the entries in “Technology_Inputs” sheet are correct and input the current design related values in “Placement_Calculator” sheet. Then use the switches provided in the “Placement_Calculator” sheet to run the macros (programs) and to view the suggested optimum distance.

5.1 Resistance Calculation of the VDDIO Power Route [Macro Name: vddResCalc]

The “VDDIO Power Route” table accepts width and number of metal routes for all metallizations in the particular technology and process (or the PDK). The vddResCalc macro accesses the resistance per unit length (micrometer) information for each of the metallizations given in the Technology_Inputs sheet and calculates the resistance offered per micrometer for each of the metal routes in the current design depending on the width and number of layers present. If no metal route of a particular metallization is present, resistance per micrometer is given as 106 Ohm in order to indicate an open circuit.

It is assumed that alternate metallizations are in parallel and adjacent metallizations are perpendicular to each other. This suggests that M1, M3, M5, M7, B1, E1, IB and MB are in parallel to each other. Similarly, M2, M4, M6, M8, B2, IA and MA are also in the same direction.

The first set of metals are perpendicular to the second set of metals. The effective resistance per micrometer of the VDDIO route is calculated from series and parallel resistances.

To illustrate the calculation of resistance per micrometer, consider a design where 20 routes of M5 are present which are 1 µm each in width. The useful information from the Technology_Inputs sheet for this calculation is shown in Table 5-2. The parameter “Wmin” refers to the minimum allowable width for the given metallization.

Table 5-2 Table for Technology_Inputs data of Metal-5

Metal Resistance								
Metal level	Wmin	Resistance (Ohm/µm)						
		Wmin	1.5Wmin	2Wmin	2.5Wmin	3Wmin	4Wmin	5Wmin
M5	0.05	9	6	4	2.25	1	0.6	0.4

The program checks to find in which category does the given width (1 µm) falls and then it finds out the integer number factor by which the given total width (= width x number of metal routes) is larger than the closest standard width given in the table. As a next step, the effective resistance per micrometer for the metal route is calculated by using the data from the Technology_Inputs sheet. This is coded as below in the Excel-macro. A comment line starts by the symbol ‘.

```
Sub vddResCalc()
‘ above line indicates the start of the macro
Dim x_mx, m5_width, m5_rails, m5_width_eff, m5_res_eff As Double
‘ other variable declarations to be included here
Sheets("Technology_Inputs").Select
‘ There are three sheets in the spreadsheet and the “Technology_Inputs” sheet needs to be selected here
x_mx = Range("C7").Value
‘ x_mx variable stores the value of Wmin, minimum width from Technology_Inputs sheet
Sheets("Placement_Calculator").Select
m5_width = Range("C22").Value
‘ m5_width variable stores the width of the M5 route
m5_rails = Range("D22").Value
‘ m5_rails variable stores the number of M5 routes in the design
m5_width_eff = (m5_width * m5_rails) / (5 * x_mx)
m5_res_eff = mx_5x / m5_width_eff
‘This is the resistance per micrometer of metal 5. The resistance of other metallizations are calculated in similar fashion
‘rest of the code
```

```
End Sub
‘ above line represents end of the macro
```

The vddResCalc macro calculates resistance per micrometer for each of the metallizations of VDDIO power route and calculates the series/parallel resistances to find the overall effective resistance per micrometer of the VDDIO power route.

5.2 Resistance Calculation of VSS Power Route [Macro Name: vssResCalc]

The “VSS Power Route” table accepts the width and the number of metal routes for all metallizations in the particular technology and process (or the PDK). The vssResCalc macro accesses the resistance per unit length (micrometer) information for each of the metallizations given in the Technology_Inputs sheet and calculates the resistance offered per micrometer for each of the metal routes in the current design depending on the width and number of layers present. If no metal route of a particular metallization is present, resistance per micrometer is given as 106 Ohm in order to indicate an open circuit. The algorithm is similar to that of vddResCalc.

The vssResCalc macro calculates resistance per micrometer for each of the metallizations of the VSS power route and applies the series/parallel resistance theorem to find out the overall effective resistance per micrometer of the VSS power route.

5.3 Resistance Calculation of Diode to Supply Route [Macro Name: diodeSupplyResCalc]

The “diodeSupplyResCalc” calculates the resistance per micrometer of the metal route which connects the diode to supply rail. This resistance is not a very significant part of the ESD network, but this is taken into account in order to make the tool more accurate. The method behind this macro is similar to “vddResCalc” and “vssResCalc”.

5.4 Resistance Calculation of PAD to Diode Route [Macro Name: padDiodeResCalc]

The “diodeSupplyResCalc” calculates the resistance per micrometer of the metal route which connects the diode to its associated PAD. This resistance is not a very significant part of the ESD network, but this is taken into account in order to make the tool more accurate. The method behind this macro is similar to “diodeSupplyResCalc”.

5.5 Resistance Calculation of HBM Up-diode [Macro Name: hbmUpDiodeResCalc]

The HBM up-diode can be an “STI diode”, a “Poly diode” or an “STI string diode”. The type of the diode can be “P+/NW”, “2STR P+/NW” or “3STR P+/NW”. The physical length of the diode can also be chosen from the given options for the particular technology. The “HBM Up-Diode” table accepts these inputs along with the perimeter of the diode used. The “hbmUpDiodeResCalc” macro accesses the technology related values of the diode from the “Technology_Inputs” sheet and calculates the effective resistance of the diode.

The steps for calculation is illustrated below for an “STI Diode” of “P+/NW” type with a length of 0.22 µm and a perimeter of 800 µm. Below is the code in the macro which executes the calculation.

```
Sub hbmUpDiodeResCalc()
‘ above line indicates the start of the macro
Dim perimeter_sti_ppnw_0p22, res_sti_ppnw_0p22, length, perimeter, res_eff As Double
Dim diode, diode_type As String
‘ other variable declarations to be included here
Sheets("Technology_Inputs").Select
perimeter_sti_ppnw_0p22 = Range("F32").Value
‘ perimeter_sti_ppnw_0p22 variable stores the standard perimeter value from Technology_Inputs sheet
Sheets("Placement_Calculator").Select
res_sti_ppnw_0p22 = Range("F35").Value
‘ res_sti_ppnw_0p22 variable stores the resistance value corresponding to the perimeter value in perimeter_sti_ppnw_0p22 variable
```

```
If diode = "STI Diode" And diode_type = "P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_ppnw_0p22 * res_sti_ppnw_0p22 / perimeter
‘ rest of the code
End Sub
‘ above line indicates the end of the macro
```

5.6 Resistance Calculation of HBM Down-diode [Macro Name: hbmDownDiodeResCalc]

The HBM down-diode can be an “STI diode”, a “Poly diode”, or an “STI string diode”. The type of the diode can be “P+/NW”, “N+/PW”, “2STR P+/NW”, or “3STR P+/NW”. The length of the diode can also be chosen from the given options for the particular technology. The “HBM Down-Diode” table accepts these inputs along with the perimeter of the diode used. The “hbmDownDiodeResCalc” macro accesses the technology related values of the diode from the “Technology_Inputs” sheet and calculates the effective resistance of the diode. The method is the same as the calculation of resistance of an HBM up-diode.

5.7 Resistance Calculation of CDM Up-diode [Macro Name: cdmUpDiodeResCalc]

A CDM up-diode can be an “STI diode”, a “Poly diode” or an “STI string diode”. The type of diode can be “P+/NW”, “2STR P+/NW” or “3STR P+/NW”. The length of the diode can also be chosen from the given options for the particular technology. “CDM Up-Diode” table accepts these inputs along with the perimeter of the diode used. The “cdmUpDiodeResCalc” macro accesses the technology related values of the diode from the “Technology_Inputs” sheet and calculates the effective resistance of the diode. The method is the same as the calculation of resistance of the HBM up-diode.

5.8 Resistance Calculation of CDM Down-diode [Macro Name: cdmDownDiodeResCalc]

CDM down-diode can be an “STI diode”, a “Poly diode” or an “STI string diode”. The type of the diode can be “P+/NW”, “N+/PW”, “2STR P+/NW” or “3STR P+/NW”. Length of the diode can also be chosen from the given options for the particular technology. “CDM Down-

Diode" table accepts these inputs along with the perimeter of the diode used. The "cdmDownDiodeResCalc" macro accesses the technology-related values of the diode from the "Technology_Inputs" sheet and calculates the effective resistance of the diode. The method is the same as the calculation of resistance of the HBM up-diode.

5.9 Resistance Calculation of RC-triggered Power Clamp [Macro Name: clampResCalc]

The RC-triggered power clamp device can be an "EG" device [transistor with a higher minimum length, an IO transistor] or an "SG" device [transistor with the lowest minimum length for the technology, a core transistor]. The length of the device can be chosen according to the type of the device used. The "RC-triggered Power Clamp" table accepts these values along with width of the device. The "clampResCalc" macro accesses the technology related values of the clamp from the "Technology_Inputs" sheet and calculates the effective resistance of the clamp. The method remains the same as that of the "hbmUpDiodeResCalc" macro.

5.10 Debug Help

The program has eight different scenarios to evaluate and report optimum distance as given in Table 5-1. The "DEBUG HELP" table is provided to give the user a better picture about the ESD network analysis. The "Vbus" column gives the maximum value of voltage which can be dropped along the power bus. Depending on the scenario, the equation to calculate Vbus has minor changes since the current flow varies as shown in Figure 4-2 and Figure 4-3. In order to illustrate the equation, consider an HBM event in which the junction-breakdown of the PMOS driver is considered. The equation used is, Vbus = Vtrig - (Von diode + Von clamp +(HBM current *(resistance of HBM down-diode + resistance of RC-triggered power clamp + resistance of pad to diode route+ resistance of diode to supply route))) . Since the current is determined according to the type of ESD event considered, Rbus (total resistance of the power bus) can be calculated using Ohm's law. The "DEBUG HELP" table is provided for both category-1 and category-2 cases as shown in Figure 5-1 and Figure 5-2 respectively.

5.11 Optimum Distance for Category-1

The optimum distance for each of the eight scenarios mentioned in section 6.10 for the category-1 type of arrangement is calculated and printed. Out of the eight distances shown, the “suggested distance” would be the least among the eight values. If the design has a power clamp (power IO) at least at the suggested distance, there is very high possibility that the design would not fail due to ESD events if the arrangement is of Category-1.

5.12 Optimum Distance for Category-2

The optimum distance for each of the eight scenarios mentioned in section 5.10 for the category-2 type of arrangement is calculated and printed. Out of the eight distances shown, the “suggested dutsnace” would be the least among the eight values. If the design has a power clamp (power IO) at least at the suggested distance, there is very high possibility that the design would not fail due to ESD events if the arrangement is of Category-2.

The “IO Planner” works for different technologies since it has a “Technology_Inputs” sheet. This sheet accepts technology related values for resistances, voltages etc and the engine reads them while performing the calculation. The “Debug help” facility helps the user to easily figure out where the margin lost is or what needs to be improved in the network.

Figure 5-6 represents an IO bank for which the “IO Planner” was used to estimate the placement of power clamps. The program suggested 400 µm as the optimum distance to place power clamps for category-2 type of arrangement.

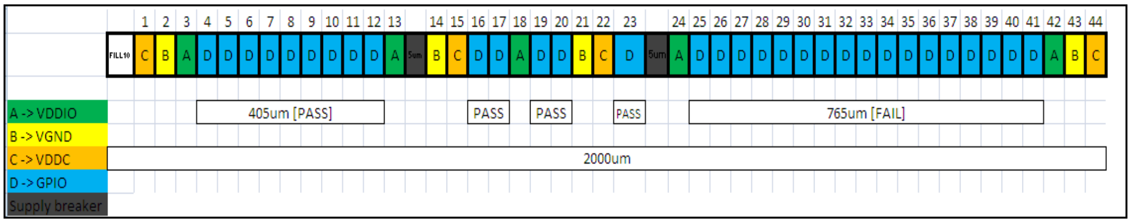


Figure 5-6 A representation of IO bank planned using the "IO Planner"

5.13 Correlation with Schematic Simulation:

The credibility of the "IO planner" tool is validated by doing an HSPICE simulation of schematic netlist of the ESD network and comparing the results with the result provded by the "IO Planner". Schematic arrangement used to do the verification is shown in Figure 5-7.

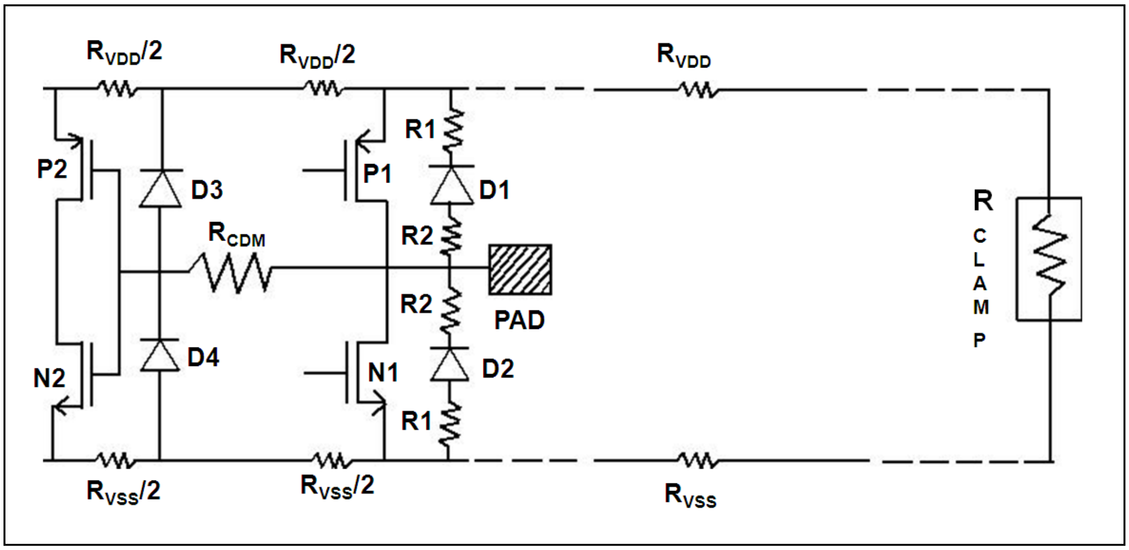


Figure 5-7 Schematic arrangement for ESD network simulation

The optimum distance suggested by the program is found to be 25% higher than the value obtained through simulation. The value suggested by the "IO planner" is more pessimistic than the simulation result. This provides an additional safety factor against ESD failure. Thus the probability for ESD failure is less than what would have been the probability with the distance obtained through simulation. In other words, "IO planner" gives a safer placement option.

Chapter 6

Conclusion and Future Scope

The layout area of the GPIO circuit is compact and is one of the smallest sizes achieved for the 28 nm technology with functions such as programmable drive strength and crowbar current control implemented.

The clamp placement calculator tool is meant to reduce the ESD failure on silicon and it has a fairly reasonable correlation with simulation results. The limitations of the tool are, it does not consider ESD protection schemes other than diode-based, parasitic capacitance and inductance is not considered for analysis. The distance suggested by the “IO Planner” is more pessimistic than the value obtained through simulation, hence the distance suggested by tool is always safer. Since the program is not running a simulation, it is very fast and gives result almost instantaneously.

As a future improvement, the tool may be developed to run an actual schematic simulation and this suggests the optimum distance to place clamps which should ideally give the user more window to place IO buffers without repeating power clamps. Current work is good for diode based protection schemes and the tool may be developed to work for other schemes in future. Also, the program can be rewritten in BASIC to make it more widely acceptable.

Appendix-A

Visual Basic Programs

```
Sub vddResCalc()

' vddResCalc Macro

' To calculate the resistance of vdd power bus

  Dim m1_1x, m1_1p5x, m1_2x, m1_2p5x, m1_3x, m1_4x, m1_5x, m1_iso As Double

  Dim mx_1x, mx_1p5x, mx_2x, mx_2p5x, mx_3x, mx_4x, mx_5x, mx_iso As Double

  Dim bx_1x, bx_1p5x, bx_2x, bx_2p5x, bx_3x, bx_4x, bx_5x, bx_iso As Double

  Dim e1_1x, e1_1p5x, e1_2x, e1_2p5x, e1_3x, e1_4x, e1_5x, e1_iso As Double

  Dim ix_1x, ix_1p5x, ix_2x, ix_2p5x, ix_3x, ix_4x, ix_5x, ix_iso As Double

  Dim ma_mb_1x, ma_mb_1p5x, ma_mb_2x, ma_mb_2p5x, ma_mb_3x, ma_mb_4x, ma_mb_5x, ma_mb_iso As Double

  Dim x_m1, x_mx, x_bx, x_e1, x_ix, x_ma_mb, width, width_eff, length As Double

  Dim m1_width_eff, m1_res_eff, mx_width_eff, mx_res_eff, bx_width_eff, bx_res_eff, e1_width_eff, e1_res_eff, ix_width_eff, ma_mb_width_eff As Double

  Dim metal As String

  Dim store As Range

  Sheets("Placement_Calculator").Select

  Sheets("Technology_Inputs").Select

  x_m1 = Range("C6").Value

  x_mx = Range("C7").Value

  x_bx = Range("C8").Value

  x_e1 = Range("C9").Value

  x_ix = Range("C10").Value

  x_ma_mb = Range("C11").Value


  m1_1x = Range("D6").Value
```

```
m1_1p5x = Range("E6").Value
m1_2x = Range("F6").Value
m1_2p5x = Range("G6").Value
m1_3x = Range("H6").Value
m1_4x = Range("I6").Value
m1_5x = Range("J6").Value
m1_iso = Range("K6").Value

mx_1x = Range("D7").Value
mx_1p5x = Range("E7").Value
mx_2x = Range("F7").Value
mx_2p5x = Range("G7").Value
mx_3x = Range("H7").Value
mx_4x = Range("I7").Value
mx_5x = Range("J7").Value
mx_iso = Range("K7").Value

bx_1x = Range("D8").Value
bx_1p5x = Range("E8").Value
bx_2x = Range("F8").Value
bx_2p5x = Range("G8").Value
bx_3x = Range("H8").Value
bx_4x = Range("I8").Value
bx_5x = Range("J8").Value
bx_iso = Range("K8").Value
```

```
e1_1x = Range("D9").Value

e1_1p5x = Range("E9").Value

e1_2x = Range("F9").Value

e1_2p5x = Range("G9").Value

e1_3x = Range("H9").Value

e1_4x = Range("I9").Value

e1_5x = Range("J9").Value

e1_iso = Range("K9").Value


ix_1x = Range("D10").Value

ix_1p5x = Range("E10").Value

ix_2x = Range("F10").Value

ix_2p5x = Range("G10").Value

ix_3x = Range("H10").Value

ix_4x = Range("I10").Value

ix_5x = Range("J10").Value

ix_iso = Range("K10").Value


ma_mb_1x = Range("D11").Value

ma_mb_1p5x = Range("E11").Value

ma_mb_2x = Range("F11").Value

ma_mb_2p5x = Range("G11").Value

ma_mb_3x = Range("H11").Value

ma_mb_4x = Range("I11").Value
```

```
  ma_mb_5x = Range("J11").Value

  ma_mb_iso = Range("K11").Value

 Sheets("Placement_Calculator").Select

 m1_width = Range("C18").Value

m1_rails = Range("D18").Value

    If m1_rails = 0 Then

    m1_res_eff = 1000000

    ElseIf m1_width < 1.5 * x_m1 Then

    m1_width_eff = 1 * x_m1 * m1_rails

    m1_res_eff = m1_1x / m1_width_eff

    ElseIf m1_width < 2 * x_m1 Then

    m1_width_eff = 1.5 * x_m1 * m1_rails

    m1_res_eff = m1_1p5x / m1_width_eff

    ElseIf m1_width < 2.5 * x_m1 Then

    m1_width_eff = 2 * x_m1 * m1_rails

    m1_res_eff = m1_2x / m1_width_eff

    ElseIf m1_width < 3 * x_m1 Then

    m1_width_eff = 2.5 * x_m1 * m1_rails

    m1_res_eff = m1_2p5x / m1_width_eff

    ElseIf m1_width < 4 * x_m1 Then

    m1_width_eff = 3 * x_m1 * m1_rails

    m1_res_eff = m1_3x / m1_width_eff

    ElseIf m1_width < 5 * x_m1 Then

    m1_width_eff = 4 * x_m1 * m1_rails

    m1_res_eff = m1_4x / m1_width_eff
    Else
    m1_width_eff = (m1_width * m1_rails) / (5 * x_m1)
    m1_res_eff = m1_5x / m1_width_eff
    End If

Range("E18") = m1_res_eff

m2_width = Range("C19").Value
m2_rails = Range("D19").Value
    If m2_rails = 0 Then
    m2_res_eff = 1000000
    ElseIf m2_width < 1.5 * x_mx Then
    m2_width_eff = 1 * x_mx * m2_rails
    m2_res_eff = mx_1x / m2_width_eff
    ElseIf m2_width < 2 * x_mx Then
    m2_width_eff = 1.5 * x_mx * m2_rails
    m2_res_eff = mx_1p5x / m2_width_eff
    ElseIf m2_width < 2.5 * x_mx Then
    m2_width_eff = 2 * x_mx * m2_rails
    m2_res_eff = mx_2x / m2_width_eff
    ElseIf m2_width < 3 * x_mx Then
    m2_width_eff = 2.5 * x_mx * m2_rails
    m2_res_eff = mx_2p5x / m2_width_eff
    ElseIf m2_width < 4 * x_mx Then
    m2_width_eff = 3 * x_mx * m2_rails
    m2_res_eff = mx_3x / m2_width_eff
    ElseIf m2_width < 5 * x_mx Then
    m2_width_eff = 4 * x_mx * m2_rails
    m2_res_eff = mx_4x / m2_width_eff
    Else
    m2_width_eff = (m2_width * m2_rails) / (5 * x_mx)
    m2_res_eff = mx_5x / m2_width_eff
    End If

Range("E19") = m2_res_eff

m3_width = Range("C20").Value
m3_rails = Range("D20").Value
    If m3_rails = 0 Then
    m3_res_eff = 1000000
    ElseIf m3_width < 1.5 * x_mx Then
    m3_width_eff = 1 * x_mx * m3_rails
    m3_res_eff = mx_1x / m3_width_eff
    ElseIf m3_width < 2 * x_mx Then
    m3_width_eff = 1.5 * x_mx * m3_rails
    m3_res_eff = mx_1p5x / m3_width_eff
    ElseIf m3_width < 2.5 * x_mx Then
    m3_width_eff = 2 * x_mx * m3_rails
    m3_res_eff = mx_2x / m3_width_eff
    ElseIf m3_width < 3 * x_mx Then
    m3_width_eff = 2.5 * x_mx * m3_rails
    m3_res_eff = mx_2p5x / m3_width_eff
    ElseIf m3_width < 4 * x_mx Then
    m3_width_eff = 3 * x_mx * m3_rails
    m3_res_eff = mx_3x / m3_width_eff
    ElseIf m3_width < 5 * x_mx Then
    m3_width_eff = 4 * x_mx * m3_rails
    m3_res_eff = mx_4x / m3_width_eff
    Else
    m3_width_eff = (m3_width * m3_rails) / (5 * x_mx)
    m3_res_eff = mx_5x / m3_width_eff
    End If

Range("E20") = m3_res_eff


m4_width = Range("C21").Value
m4_rails = Range("D21").Value
    If m4_rails = 0 Then
    m4_res_eff = 1000000
    ElseIf m4_width < 1.5 * x_mx Then
    m4_width_eff = 1 * x_mx * m4_rails
    m4_res_eff = mx_1x / m4_width_eff
    ElseIf m4_width < 2 * x_mx Then
    m4_width_eff = 1.5 * x_mx * m4_rails
    m4_res_eff = mx_1p5x / m4_width_eff
    ElseIf m4_width < 2.5 * x_mx Then

    m4_width_eff = 2 * x_mx * m4_rails

    m4_res_eff = mx_2x / m4_width_eff

    ElseIf m4_width < 3 * x_mx Then

    m4_width_eff = 2.5 * x_mx * m4_rails

    m4_res_eff = mx_2p5x / m4_width_eff

    ElseIf m4_width < 4 * x_mx Then

    m4_width_eff = 3 * x_mx * m4_rails

    m4_res_eff = mx_3x / m4_width_eff

    ElseIf m4_width < 5 * x_mx Then

    m4_width_eff = 4 * x_mx * m4_rails

    m4_res_eff = mx_4x / m4_width_eff

    Else

    m4_width_eff = (m4_width * m4_rails) / (5 * x_mx)

    m4_res_eff = mx_5x / m4_width_eff

    End If


Range("E21") = m4_res_eff


m5_width = Range("C22").Value

m5_rails = Range("D22").Value

    If m5_rails = 0 Then

    m5_res_eff = 1000000

    ElseIf m5_width < 1.5 * x_mx Then

    m5_width_eff = 1 * x_mx * m5_rails
    m5_res_eff = mx_1x / m5_width_eff
    ElseIf m5_width < 2 * x_mx Then
    m5_width_eff = 1.5 * x_mx * m5_rails
    m5_res_eff = mx_1p5x / m5_width_eff
    ElseIf m5_width < 2.5 * x_mx Then
    m5_width_eff = 2 * x_mx * m5_rails
    m5_res_eff = mx_2x / m5_width_eff
    ElseIf m5_width < 3 * x_mx Then
    m5_width_eff = 2.5 * x_mx * m5_rails
    m5_res_eff = mx_2p5x / m5_width_eff
    ElseIf m5_width < 4 * x_mx Then
    m5_width_eff = 3 * x_mx * m5_rails
    m5_res_eff = mx_3x / m5_width_eff
    ElseIf m5_width < 5 * x_mx Then
    m5_width_eff = 4 * x_mx * m5_rails
    m5_res_eff = mx_4x / m5_width_eff
    Else
    m5_width_eff = (m5_width * m5_rails) / (5 * x_mx)
    m5_res_eff = mx_5x / m5_width_eff
    End If

Range("E22") = m5_res_eff


m6_width = Range("C23").Value
m6_rails = Range("D23").Value
```

```
If m6_rails = 0 Then
m6_res_eff = 1000000
ElseIf m6_width < 1.5 * x_mx Then
m6_width_eff = 1 * x_mx * m6_rails
m6_res_eff = mx_1x / m6_width_eff
ElseIf m6_width < 2 * x_mx Then
m6_width_eff = 1.5 * x_mx * m6_rails
m6_res_eff = mx_1p5x / m6_width_eff
ElseIf m6_width < 2.5 * x_mx Then
m6_width_eff = 2 * x_mx * m6_rails
m6_res_eff = mx_2x / m6_width_eff
ElseIf m6_width < 3 * x_mx Then
m6_width_eff = 2.5 * x_mx * m6_rails
m6_res_eff = mx_2p5x / m6_width_eff
ElseIf m6_width < 4 * x_mx Then
m6_width_eff = 3 * x_mx * m6_rails
m6_res_eff = mx_3x / m6_width_eff
ElseIf m6_width < 5 * x_mx Then
m6_width_eff = 4 * x_mx * m6_rails
m6_res_eff = mx_4x / m6_width_eff
Else
m6_width_eff = (m6_width * m6_rails) / (5 * x_mx)
m6_res_eff = mx_5x / m6_width_eff
End If
```

```
Range("E23") = m6_res_eff


m7_width = Range("C24").Value
m7_rails = Range("D24").Value
   If m7_rails = 0 Then
   m7_res_eff = 1000000
   ElseIf m7_width < 1.5 * x_mx Then
   m7_width_eff = 1 * x_mx * m7_rails
   m7_res_eff = mx_1x / m7_width_eff
   ElseIf m7_width < 2 * x_mx Then
   m7_width_eff = 1.5 * x_mx * m7_rails
   m7_res_eff = mx_1p5x / m7_width_eff
   ElseIf m7_width < 2.5 * x_mx Then
   m7_width_eff = 2 * x_mx * m7_rails
   m7_res_eff = mx_2x / m7_width_eff
   ElseIf m7_width < 3 * x_mx Then
   m7_width_eff = 2.5 * x_mx * m7_rails
   m7_res_eff = mx_2p5x / m7_width_eff
   ElseIf m7_width < 4 * x_mx Then
   m7_width_eff = 3 * x_mx * m7_rails
   m7_res_eff = mx_3x / m7_width_eff
   ElseIf m7_width < 5 * x_mx Then
   m7_width_eff = 4 * x_mx * m7_rails
   m7_res_eff = mx_4x / m7_width_eff
   Else
```

```
    m7_width_eff = (m7_width * m7_rails) / (5 * x_mx)

    m7_res_eff = mx_5x / m7_width_eff

    End If


Range("E24") = m7_res_eff

m8_width = Range("C25").Value

m8_rails = Range("D25").Value

    If m8_rails = 0 Then

    m8_res_eff = 1000000

    ElseIf m7_width < 1.5 * x_mx Then

    m8_width_eff = 1 * x_mx * m8_rails

    m8_res_eff = mx_1x / m8_width_eff

    ElseIf m7_width < 2 * x_mx Then

    m8_width_eff = 1.5 * x_mx * m8_rails

    m8_res_eff = mx_1p5x / m8_width_eff

    ElseIf m7_width < 2.5 * x_mx Then

    m8_width_eff = 2 * x_mx * m8_rails

    m8_res_eff = mx_2x / m8_width_eff

    ElseIf m7_width < 3 * x_mx Then

    m8_width_eff = 2.5 * x_mx * m8_rails

    m8_res_eff = mx_2p5x / m8_width_eff

    ElseIf m7_width < 4 * x_mx Then

    m8_width_eff = 3 * x_mx * m8_rails

    m8_res_eff = mx_3x / m8_width_eff

    ElseIf m7_width < 5 * x_mx Then
    m8_width_eff = 4 * x_mx * m8_rails
    m8_res_eff = mx_4x / m8_width_eff
    Else
    m8_width_eff = (m8_width * m8_rails) / (5 * x_mx)
    m8_res_eff = mx_5x / m8_width_eff
    End If
    Range("E25") = m8_res_eff


b1_width = Range("C26").Value
b1_rails = Range("D26").Value
    If b1_rails = 0 Then
    b1_res_eff = 1000000
    ElseIf b1_width < 1.5 * x_bx Then
    b1_width_eff = 1 * x_bx * b1_rails
    b1_res_eff = bx_1x
    ElseIf b1_width < 2 * x_bx Then
    b1_width_eff = 1.5 * x_bx * b1_rails
    b1_res_eff = bx_1p5x
    ElseIf b1_width < 2.5 * x_bx Then
    b1_width_eff = 2 * x_bx * b1_rails
    b1_res_eff = bx_2x
    ElseIf b1_width < 3 * x_bx Then
    b1_width_eff = 2.5 * x_bx * b1_rails
    b1_res_eff = bx_2p5x
    ElseIf b1_width < 4 * x_bx Then
    b1_width_eff = 3 * x_bx * b1_rails
    b1_res_eff = bx_3x
    ElseIf b1_width < 5 * x_bx Then
    b1_width_eff = 4 * x_bx * b1_rails
    b1_res_eff = bx_4x
    Else
    b1_width_eff = (b1_width * b1_rails) / (5 * x_bx)
    b1_res_eff = bx_5x / b1_width_eff
    End If

Range("E26") = b1_res_eff
b2_width = Range("C27").Value
b2_rails = Range("D27").Value
    If b2_rails = 0 Then
    b2_res_eff = 1000000
    ElseIf b2_width < 1.5 * x_bx Then
    b2_width_eff = 1 * x_bx * b2_rails
    b2_res_eff = bx_1x
    ElseIf b2_width < 2 * x_bx Then
    b2_width_eff = 1.5 * x_bx * b2_rails
    b2_res_eff = bx_1p5x
    ElseIf b2_width < 2.5 * x_bx Then
    b2_width_eff = 2 * x_bx * b2_rails
    b2_res_eff = bx_2x
    ElseIf b2_width < 3 * x_bx Then
    b2_width_eff = 2.5 * x_bx * b2_rails

    b2_res_eff = bx_2p5x

    ElseIf b2_width < 4 * x_bx Then

    b2_width_eff = 3 * x_bx * b2_rails

    b2_res_eff = bx_3x

    ElseIf b2_width < 5 * x_bx Then

    b2_width_eff = 4 * x_bx * b2_rails

    b2_res_eff = bx_4x

    Else

    b2_width_eff = (b2_width * b2_rails) / (5 * x_bx)

    b2_res_eff = bx_5x / b2_width_eff

    End If

    Range("E27") = b2_res_eff


e1_width = Range("C28").Value

e1_rails = Range("D28").Value

    If e1_rails = 0 Then

    e1_res_eff = 1000000

    ElseIf e1_width < 1.5 * x_e1 Then

    e1_width_eff = 1 * x_e1 * e1_rails

    e1_res_eff = e1_1x

    ElseIf e1_width < 2 * x_e1 Then

    e1_width_eff = 1.5 * x_e1 * e1_rails

    e1_res_eff = e1_1p5x

    ElseIf e1_width < 2.5 * x_e1 Then
    e1_width_eff = 2 * x_e1 * e1_rails
    e1_res_eff = e1_2x
    ElseIf e1_width < 3 * x_e1 Then
    e1_width_eff = 2.5 * x_e1 * e1_rails
    e1_res_eff = e1_2p5x
    ElseIf e1_width < 4 * x_e1 Then
    e1_width_eff = 3 * x_e1 * e1_rails
    e1_res_eff = e1_3x
    ElseIf e1_width < 5 * x_e1 Then
    e1_width_eff = 4 * x_e1 * e1_rails
    e1_res_eff = e1_4x
    Else
    e1_width_eff = (e1_width * e1_rails) / (5 * x_e1)
    e1_res_eff = e1_5x / e1_width_eff
    End If

Range("E28") = e1_res_eff
ia_width = Range("C29").Value
ia_rails = Range("D29").Value
    If ia_rails = 0 Then
    ia_res_eff = 1000000
    ElseIf ia_width < 1.5 * x_ix Then
    ia_width_eff = 1 * x_ix * ia_rails
    ia_res_eff = ix_1x
    ElseIf ia_width < 2 * x_ix Then
    ia_width_eff = 1.5 * x_ix * ia_rails

    ia_res_eff = ix_1p5x

    ElseIf ia_width < 2.5 * x_ix Then

    ia_width_eff = 2 * x_ix * ia_rails

    ia_res_eff = ix_2x

    ElseIf ia_width < 3 * x_ix Then

    ia_width_eff = 2.5 * x_ix * ia_rails

    ia_res_eff = ix_2p5x

    ElseIf ia_width < 4 * x_ix Then

    ia_width_eff = 3 * x_ix * ia_rails

    ia_res_eff = ix_3x

    ElseIf ia_width < 5 * x_ix Then

    ia_width_eff = 4 * x_ix * ia_rails

    ia_res_eff = ix_4x

    Else

    ia_width_eff = (ia_width * ia_rails) / (5 * x_ix)

    ia_res_eff = ix_5x / ia_width_eff

    End If


Range("E29") = ia_res_eff

ib_width = Range("C30").Value

ib_rails = Range("D30").Value

    If ib_rails = 0 Then

    ib_res_eff = 1000000

    ElseIf ib_width < 1.5 * x_ix Then
    ib_width_eff = 1 * x_ix * ib_rails

    ib_res_eff = ix_1x

    ElseIf ib_width < 2 * x_ix Then

    ib_width_eff = 1.5 * x_ix * ib_rails

    ib_res_eff = ix_1p5x

    ElseIf ib_width < 2.5 * x_ix Then

    ib_width_eff = 2 * x_ix * ib_rails

    ib_res_eff = ix_2x

    ElseIf ib_width < 3 * x_ix Then

    ib_width_eff = 2.5 * x_ix * ib_rails

    ib_res_eff = ix_2p5x

    ElseIf ib_width < 4 * x_ix Then

    ib_width_eff = 3 * x_ix * ib_rails

    ib_res_eff = ix_3x

    ElseIf ib_width < 5 * x_ix Then

    ib_width_eff = 4 * x_ix * ib_rails

    ib_res_eff = ix_4x

    Else

    ib_width_eff = (ib_width * ib_rails) / (5 * x_ix)

    ib_res_eff = ix_5x / ib_width_eff

    End If

    Range("E30") = ia_res_eff


ma_width = Range("C31").Value

ma_rails = Range("D31").Value
```

```
If ma_rails = 0 Then
ma_res_eff = 1000000
ElseIf ma_width < 1.5 * x_mx Then
ma_width_eff = 1 * x_mx * ma_rails
ma_res_eff = mx_1x
ElseIf ma_width < 2 * x_mx Then
ma_width_eff = 1.5 * x_mx * ma_rails
ma_res_eff = mx_1p5x
ElseIf ma_width < 2.5 * x_mx Then
ma_width_eff = 2 * x_mx * ma_rails
ma_res_eff = mx_2x
ElseIf ma_width < 3 * x_mx Then
ma_width_eff = 2.5 * x_mx * ma_rails
ma_res_eff = mx_2p5x
ElseIf ma_width < 4 * x_mx Then
ma_width_eff = 3 * x_mx * ma_rails
ma_res_eff = mx_3x
ElseIf ma_width < 5 * x_mx Then
ma_width_eff = 4 * x_mx * ma_rails
ma_res_eff = mx_4x
Else
ma_width_eff = (ma_width * ma_rails) / (5 * x_mx)
ma_res_eff = mx_5x / ma_width_eff
End If
```

```
Range("E31") = ma_res_eff
mb_width = Range("C32").Value
mb_rails = Range("D32").Value
    If mb_rails = 0 Then
    mb_res_eff = 1000000
    ElseIf mb_width < 1.5 * x_mx Then
    mb_width_eff = 1 * x_mx * mb_rails
    mb_res_eff = mx_1x
    ElseIf mb_width < 2 * x_mx Then
    mb_width_eff = 1.5 * x_mx * mb_rails
    mb_res_eff = mx_1p5x
    ElseIf mb_width < 2.5 * x_mx Then
    mb_width_eff = 2 * x_mx * mb_rails
    mb_res_eff = mx_2x
    ElseIf mb_width < 3 * x_mx Then
    mb_width_eff = 2.5 * x_mx * mb_rails
    mb_res_eff = mx_2p5x
    ElseIf mb_width < 4 * x_mx Then
    mb_width_eff = 3 * x_mx * mb_rails
    mb_res_eff = mx_3x
    ElseIf mb_width < 5 * x_mx Then
    mb_width_eff = 4 * x_mx * mb_rails
    mb_res_eff = mx_4x
    Else
    mb_width_eff = (mb_width * mb_rails) / (5 * x_mx)
    mb_res_eff = mx_5x / mb_width_eff
    End If
    Range("E32") = mb_res_eff

hor_res_eff = 1 / ((1 / m2_res_eff) + (1 / m4_res_eff) + (1 / m6_res_eff) + (1 / m8_res_eff) + (1 /
b2_res_eff) + (1 / ia_res_eff) + (1 / ma_res_eff))
vert_res_eff = 1 / ((1 / m1_res_eff) + (1 / m3_res_eff) + (1 / m5_res_eff) + (1 / m7_res_eff) + (1 /
b1_res_eff) + (1 / ib_res_eff) + (1 / mb_res_eff))
numerator = ((2 * vert_res_eff) + hor_res_eff) * hor_res_eff
denominator = 2 * (vert_res_eff + hor_res_eff)
res_eff = numerator / denominator

Range("F18") = res_eff

End Sub



Sub vssResCalc()
' vddResCalc Macro
' To calculate the resistance of vdd power bus

  Dim m1_1x, m1_1p5x, m1_2x, m1_2p5x, m1_3x, m1_4x, m1_5x, m1_iso As Double
  Dim mx_1x, mx_1p5x, mx_2x, mx_2p5x, mx_3x, mx_4x, mx_5x, mx_iso As Double
  Dim bx_1x, bx_1p5x, bx_2x, bx_2p5x, bx_3x, bx_4x, bx_5x, bx_iso As Double
  Dim e1_1x, e1_1p5x, e1_2x, e1_2p5x, e1_3x, e1_4x, e1_5x, e1_iso As Double
  Dim ix_1x, ix_1p5x, ix_2x, ix_2p5x, ix_3x, ix_4x, ix_5x, ix_iso As Double
  Dim ma_mb_1x, ma_mb_1p5x, ma_mb_2x, ma_mb_2p5x, ma_mb_3x, ma_mb_4x, ma_mb_5x, ma_mb_iso As Double

  Dim x_m1, x_mx, x_bx, x_e1, x_ix, x_ma_mb, width, width_eff, length As Double
  Dim m1_width_eff, m1_res_eff, mx_width_eff, mx_res_eff, bx_width_eff, bx_res_eff, e1_width_eff, e1_res_eff, ix_width_eff, ma_mb_width_eff As Double
  Dim metal As String
  Dim store As Range

Sheets("Placement_Calculator").Select

  'width = Range("B10").Value
  'metal = Range("B9").Value

Sheets("Technology_Inputs").Select

  x_m1 = Range("C6").Value
  x_mx = Range("C7").Value
  x_bx = Range("C8").Value
  x_e1 = Range("C9").Value
  x_ix = Range("C10").Value
  x_ma_mb = Range("C11").Value
```

```
m1_1x = Range("D6").Value

m1_1p5x = Range("E6").Value

m1_2x = Range("F6").Value

m1_2p5x = Range("G6").Value

m1_3x = Range("H6").Value

m1_4x = Range("I6").Value

m1_5x = Range("J6").Value

m1_iso = Range("K6").Value


mx_1x = Range("D7").Value

mx_1p5x = Range("E7").Value

mx_2x = Range("F7").Value

mx_2p5x = Range("G7").Value

mx_3x = Range("H7").Value

mx_4x = Range("I7").Value

mx_5x = Range("J7").Value

mx_iso = Range("K7").Value


bx_1x = Range("D8").Value

bx_1p5x = Range("E8").Value

bx_2x = Range("F8").Value

bx_2p5x = Range("G8").Value

bx_3x = Range("H8").Value

bx_4x = Range("I8").Value

bx_5x = Range("J8").Value
```

```
bx_iso = Range("K8").Value

e1_1x = Range("D9").Value
e1_1p5x = Range("E9").Value
e1_2x = Range("F9").Value
e1_2p5x = Range("G9").Value
e1_3x = Range("H9").Value
e1_4x = Range("I9").Value
e1_5x = Range("J9").Value
e1_iso = Range("K9").Value

ix_1x = Range("D10").Value
ix_1p5x = Range("E10").Value
ix_2x = Range("F10").Value
ix_2p5x = Range("G10").Value
ix_3x = Range("H10").Value
ix_4x = Range("I10").Value
ix_5x = Range("J10").Value
ix_iso = Range("K10").Value

ma_mb_1x = Range("D11").Value
ma_mb_1p5x = Range("E11").Value
ma_mb_2x = Range("F11").Value
ma_mb_2p5x = Range("G11").Value
ma_mb_3x = Range("H11").Value
```

```
 ma_mb_4x = Range("I11").Value

 ma_mb_5x = Range("J11").Value

 ma_mb_iso = Range("K11").Value


Sheets("Placement_Calculator").Select


m1_width = Range("C38").Value

m1_rails = Range("D38").Value

  If m1_rails = 0 Then

  m1_res_eff = 1000000

  ElseIf m1_width < 1.5 * x_m1 Then

  m1_width_eff = 1 * x_m1 * m1_rails

  m1_res_eff = m1_1x / m1_width_eff

  ElseIf m1_width < 2 * x_m1 Then

  m1_width_eff = 1.5 * x_m1 * m1_rails

  m1_res_eff = m1_1p5x / m1_width_eff

  ElseIf m1_width < 2.5 * x_m1 Then

  m1_width_eff = 2 * x_m1 * m1_rails

  m1_res_eff = m1_2x / m1_width_eff

  ElseIf m1_width < 3 * x_m1 Then

  m1_width_eff = 2.5 * x_m1 * m1_rails

  m1_res_eff = m1_2p5x / m1_width_eff

  ElseIf m1_width < 4 * x_m1 Then

  m1_width_eff = 3 * x_m1 * m1_rails

  m1_res_eff = m1_3x / m1_width_eff
```

```
    ElseIf m1_width < 5 * x_m1 Then
    m1_width_eff = 4 * x_m1 * m1_rails
    m1_res_eff = m1_4x / m1_width_eff
    Else
    m1_width_eff = (m1_width * m1_rails) / (5 * x_m1)
    m1_res_eff = m1_5x / m1_width_eff
    End If

Range("E38") = m1_res_eff


m2_width = Range("C39").Value
m2_rails = Range("D39").Value
    If m2_rails = 0 Then
    m2_res_eff = 1000000
    ElseIf m2_width < 1.5 * x_mx Then
    m2_width_eff = 1 * x_mx * m2_rails
    m2_res_eff = mx_1x / m2_width_eff
    ElseIf m2_width < 2 * x_mx Then
    m2_width_eff = 1.5 * x_mx * m2_rails
    m2_res_eff = mx_1p5x / m2_width_eff
    ElseIf m2_width < 2.5 * x_mx Then
    m2_width_eff = 2 * x_mx * m2_rails
    m2_res_eff = mx_2x / m2_width_eff
    ElseIf m2_width < 3 * x_mx Then
    m2_width_eff = 2.5 * x_mx * m2_rails
    m2_res_eff = mx_2p5x / m2_width_eff

    ElseIf m2_width < 4 * x_mx Then

    m2_width_eff = 3 * x_mx * m2_rails

    m2_res_eff = mx_3x / m2_width_eff

    ElseIf m2_width < 5 * x_mx Then

    m2_width_eff = 4 * x_mx * m2_rails

    m2_res_eff = mx_4x / m2_width_eff

    Else

    m2_width_eff = (m2_width * m2_rails) / (5 * x_mx)

    m2_res_eff = mx_5x / m2_width_eff

    End If


Range("E39") = m2_res_eff


m3_width = Range("C40").Value

m3_rails = Range("D40").Value

    If m3_rails = 0 Then

    m3_res_eff = 1000000

    ElseIf m3_width < 1.5 * x_mx Then

    m3_width_eff = 1 * x_mx * m3_rails

    m3_res_eff = mx_1x / m3_width_eff

    ElseIf m3_width < 2 * x_mx Then

    m3_width_eff = 1.5 * x_mx * m3_rails

    m3_res_eff = mx_1p5x / m3_width_eff

    ElseIf m3_width < 2.5 * x_mx Then
    m3_width_eff = 2 * x_mx * m3_rails

    m3_res_eff = mx_2x / m3_width_eff

    ElseIf m3_width < 3 * x_mx Then

    m3_width_eff = 2.5 * x_mx * m3_rails

    m3_res_eff = mx_2p5x / m3_width_eff

    ElseIf m3_width < 4 * x_mx Then

    m3_width_eff = 3 * x_mx * m3_rails

    m3_res_eff = mx_3x / m3_width_eff

    ElseIf m3_width < 5 * x_mx Then

    m3_width_eff = 4 * x_mx * m3_rails

    m3_res_eff = mx_4x / m3_width_eff

    Else

    m3_width_eff = (m3_width * m3_rails) / (5 * x_mx)

    m3_res_eff = mx_5x / m3_width_eff

    End If


Range("E40") = m3_res_eff


m4_width = Range("C41").Value

m4_rails = Range("D41").Value

    If m4_rails = 0 Then

    m4_res_eff = 1000000

    ElseIf m4_width < 1.5 * x_mx Then

    m4_width_eff = 1 * x_mx * m4_rails

    m4_res_eff = mx_1x / m4_width_eff
    ElseIf m4_width < 2 * x_mx Then

    m4_width_eff = 1.5 * x_mx * m4_rails

    m4_res_eff = mx_1p5x / m4_width_eff

    ElseIf m4_width < 2.5 * x_mx Then

    m4_width_eff = 2 * x_mx * m4_rails

    m4_res_eff = mx_2x / m4_width_eff

    ElseIf m4_width < 3 * x_mx Then

    m4_width_eff = 2.5 * x_mx * m4_rails

    m4_res_eff = mx_2p5x / m4_width_eff

    ElseIf m4_width < 4 * x_mx Then

    m4_width_eff = 3 * x_mx * m4_rails

    m4_res_eff = mx_3x / m4_width_eff

    ElseIf m4_width < 5 * x_mx Then

    m4_width_eff = 4 * x_mx * m4_rails

    m4_res_eff = mx_4x / m4_width_eff

    Else

    m4_width_eff = (m4_width * m4_rails) / (5 * x_mx)

    m4_res_eff = mx_5x / m4_width_eff

    End If


Range("E41") = m4_res_eff


m5_width = Range("C42").Value

m5_rails = Range("D42").Value

    If m5_rails = 0 Then
    m5_res_eff = 1000000

    ElseIf m5_width < 1.5 * x_mx Then

    m5_width_eff = 1 * x_mx * m5_rails

    m5_res_eff = mx_1x / m5_width_eff

    ElseIf m5_width < 2 * x_mx Then

    m5_width_eff = 1.5 * x_mx * m5_rails

    m5_res_eff = mx_1p5x / m5_width_eff

    ElseIf m5_width < 2.5 * x_mx Then

    m5_width_eff = 2 * x_mx * m5_rails

    m5_res_eff = mx_2x / m5_width_eff

    ElseIf m5_width < 3 * x_mx Then

    m5_width_eff = 2.5 * x_mx * m5_rails

    m5_res_eff = mx_2p5x / m5_width_eff

    ElseIf m5_width < 4 * x_mx Then

    m5_width_eff = 3 * x_mx * m5_rails

    m5_res_eff = mx_3x / m5_width_eff

    ElseIf m5_width < 5 * x_mx Then

    m5_width_eff = 4 * x_mx * m5_rails

    m5_res_eff = mx_4x / m5_width_eff

    Else

    m5_width_eff = (m5_width * m5_rails) / (5 * x_mx)

    m5_res_eff = mx_5x / m5_width_eff

    End If


Range("E42") = m5_res_eff
```

```
m6_width = Range("C43").Value

m6_rails = Range("D43").Value

    If m6_rails = 0 Then

    m6_res_eff = 1000000

    ElseIf m6_width < 1.5 * x_mx Then

    m6_width_eff = 1 * x_mx * m6_rails

    m6_res_eff = mx_1x / m6_width_eff

    ElseIf m6_width < 2 * x_mx Then

    m6_width_eff = 1.5 * x_mx * m6_rails

    m6_res_eff = mx_1p5x / m6_width_eff

    ElseIf m6_width < 2.5 * x_mx Then

    m6_width_eff = 2 * x_mx * m6_rails

    m6_res_eff = mx_2x / m6_width_eff

    ElseIf m6_width < 3 * x_mx Then

    m6_width_eff = 2.5 * x_mx * m6_rails

    m6_res_eff = mx_2p5x / m6_width_eff

    ElseIf m6_width < 4 * x_mx Then

    m6_width_eff = 3 * x_mx * m6_rails

    m6_res_eff = mx_3x / m6_width_eff

    ElseIf m6_width < 5 * x_mx Then

    m6_width_eff = 4 * x_mx * m6_rails

    m6_res_eff = mx_4x / m6_width_eff

    Else

    m6_width_eff = (m6_width * m6_rails) / (5 * x_mx)
    m6_res_eff = mx_5x / m6_width_eff
    End If

Range("E43") = m6_res_eff

m7_width = Range("C44").Value
m7_rails = Range("D44").Value
    If m7_rails = 0 Then
    m7_res_eff = 1000000
    ElseIf m7_width < 1.5 * x_mx Then
    m7_width_eff = 1 * x_mx * m7_rails
    m7_res_eff = mx_1x / m7_width_eff
    ElseIf m7_width < 2 * x_mx Then
    m7_width_eff = 1.5 * x_mx * m7_rails
    m7_res_eff = mx_1p5x / m7_width_eff
    ElseIf m7_width < 2.5 * x_mx Then
    m7_width_eff = 2 * x_mx * m7_rails
    m7_res_eff = mx_2x / m7_width_eff
    ElseIf m7_width < 3 * x_mx Then
    m7_width_eff = 2.5 * x_mx * m7_rails
    m7_res_eff = mx_2p5x / m7_width_eff
    ElseIf m7_width < 4 * x_mx Then
    m7_width_eff = 3 * x_mx * m7_rails
    m7_res_eff = mx_3x / m7_width_eff
    ElseIf m7_width < 5 * x_mx Then
    m7_width_eff = 4 * x_mx * m7_rails
    m7_res_eff = mx_4x / m7_width_eff
    Else
    m7_width_eff = (m7_width * m7_rails) / (5 * x_mx)
    m7_res_eff = mx_5x / m7_width_eff
    End If

Range("E44") = m7_res_eff

m8_width = Range("C45").Value
m8_rails = Range("D45").Value
    If m8_rails = 0 Then
    m8_res_eff = 1000000
    ElseIf m7_width < 1.5 * x_mx Then
    m8_width_eff = 1 * x_mx * m8_rails
    m8_res_eff = mx_1x / m8_width_eff
    ElseIf m7_width < 2 * x_mx Then
    m8_width_eff = 1.5 * x_mx * m8_rails
    m8_res_eff = mx_1p5x / m8_width_eff
    ElseIf m7_width < 2.5 * x_mx Then
    m8_width_eff = 2 * x_mx * m8_rails
    m8_res_eff = mx_2x / m8_width_eff
    ElseIf m7_width < 3 * x_mx Then
    m8_width_eff = 2.5 * x_mx * m8_rails
    m8_res_eff = mx_2p5x / m8_width_eff
    ElseIf m7_width < 4 * x_mx Then
    m8_width_eff = 3 * x_mx * m8_rails
    m8_res_eff = mx_3x / m8_width_eff
    ElseIf m7_width < 5 * x_mx Then
    m8_width_eff = 4 * x_mx * m8_rails
    m8_res_eff = mx_4x / m8_width_eff
    Else
    m8_width_eff = (m8_width * m8_rails) / (5 * x_mx)
    m8_res_eff = mx_5x / m8_width_eff
    End If

Range("E45") = m8_res_eff

b1_width = Range("C46").Value
b1_rails = Range("D46").Value
    If b1_rails = 0 Then
    b1_res_eff = 1000000
    ElseIf b1_width < 1.5 * x_bx Then
    b1_width_eff = 1 * x_bx * b1_rails
    b1_res_eff = bx_1x
    ElseIf b1_width < 2 * x_bx Then
    b1_width_eff = 1.5 * x_bx * b1_rails
    b1_res_eff = bx_1p5x
    ElseIf b1_width < 2.5 * x_bx Then
    b1_width_eff = 2 * x_bx * b1_rails
    b1_res_eff = bx_2x
    ElseIf b1_width < 3 * x_bx Then
    b1_width_eff = 2.5 * x_bx * b1_rails
    b1_res_eff = bx_2p5x
    ElseIf b1_width < 4 * x_bx Then
    b1_width_eff = 3 * x_bx * b1_rails
    b1_res_eff = bx_3x
    ElseIf b1_width < 5 * x_bx Then
    b1_width_eff = 4 * x_bx * b1_rails
    b1_res_eff = bx_4x
    Else
    b1_width_eff = (b1_width * b1_rails) / (5 * x_bx)
    b1_res_eff = bx_5x / b1_width_eff
    End If

Range("E46") = b1_res_eff


b2_width = Range("C47").Value
b2_rails = Range("D47").Value
    If b2_rails = 0 Then
    b2_res_eff = 1000000
    ElseIf b2_width < 1.5 * x_bx Then
    b2_width_eff = 1 * x_bx * b2_rails
    b2_res_eff = bx_1x
    ElseIf b2_width < 2 * x_bx Then
    b2_width_eff = 1.5 * x_bx * b2_rails

    b2_res_eff = bx_1p5x

    ElseIf b2_width < 2.5 * x_bx Then

    b2_width_eff = 2 * x_bx * b2_rails

    b2_res_eff = bx_2x

    ElseIf b2_width < 3 * x_bx Then

    b2_width_eff = 2.5 * x_bx * b2_rails

    b2_res_eff = bx_2p5x

    ElseIf b2_width < 4 * x_bx Then

    b2_width_eff = 3 * x_bx * b2_rails

    b2_res_eff = bx_3x

    ElseIf b2_width < 5 * x_bx Then

    b2_width_eff = 4 * x_bx * b2_rails

    b2_res_eff = bx_4x

    Else

    b2_width_eff = (b2_width * b2_rails) / (5 * x_bx)

    b2_res_eff = bx_5x / b2_width_eff

    End If


Range("E47") = b2_res_eff


e1_width = Range("C48").Value

e1_rails = Range("D48").Value

    If e1_rails = 0 Then

    e1_res_eff = 1000000
    ElseIf e1_width < 1.5 * x_e1 Then
    e1_width_eff = 1 * x_e1 * e1_rails
    e1_res_eff = e1_1x
    ElseIf e1_width < 2 * x_e1 Then
    e1_width_eff = 1.5 * x_e1 * e1_rails
    e1_res_eff = e1_1p5x
    ElseIf e1_width < 2.5 * x_e1 Then
    e1_width_eff = 2 * x_e1 * e1_rails
    e1_res_eff = e1_2x
    ElseIf e1_width < 3 * x_e1 Then
    e1_width_eff = 2.5 * x_e1 * e1_rails
    e1_res_eff = e1_2p5x
    ElseIf e1_width < 4 * x_e1 Then
    e1_width_eff = 3 * x_e1 * e1_rails
    e1_res_eff = e1_3x
    ElseIf e1_width < 5 * x_e1 Then
    e1_width_eff = 4 * x_e1 * e1_rails
    e1_res_eff = e1_4x
    Else
    e1_width_eff = (e1_width * e1_rails) / (5 * x_e1)
    e1_res_eff = e1_5x / e1_width_eff
    End If

Range("E48") = e1_res_eff
```

```
ia_width = Range("C49").Value
ia_rails = Range("D49").Value
    If ia_rails = 0 Then
    ia_res_eff = 1000000
    ElseIf ia_width < 1.5 * x_ix Then
    ia_width_eff = 1 * x_ix * ia_rails
    ia_res_eff = ix_1x
    ElseIf ia_width < 2 * x_ix Then
    ia_width_eff = 1.5 * x_ix * ia_rails
    ia_res_eff = ix_1p5x
    ElseIf ia_width < 2.5 * x_ix Then
    ia_width_eff = 2 * x_ix * ia_rails
    ia_res_eff = ix_2x
    ElseIf ia_width < 3 * x_ix Then
    ia_width_eff = 2.5 * x_ix * ia_rails
    ia_res_eff = ix_2p5x
    ElseIf ia_width < 4 * x_ix Then
    ia_width_eff = 3 * x_ix * ia_rails
    ia_res_eff = ix_3x
    ElseIf ia_width < 5 * x_ix Then
    ia_width_eff = 4 * x_ix * ia_rails
    ia_res_eff = ix_4x
    Else
    ia_width_eff = (ia_width * ia_rails) / (5 * x_ix)
    ia_res_eff = ix_5x / ia_width_eff
    End If

Range("E49") = ia_res_eff

ib_width = Range("C50").Value
ib_rails = Range("D50").Value
    If ib_rails = 0 Then
    ib_res_eff = 1000000
    ElseIf ib_width < 1.5 * x_ix Then
    ib_width_eff = 1 * x_ix * ib_rails
    ib_res_eff = ix_1x
    ElseIf ib_width < 2 * x_ix Then
    ib_width_eff = 1.5 * x_ix * ib_rails
    ib_res_eff = ix_1p5x
    ElseIf ib_width < 2.5 * x_ix Then
    ib_width_eff = 2 * x_ix * ib_rails
    ib_res_eff = ix_2x
    ElseIf ib_width < 3 * x_ix Then
    ib_width_eff = 2.5 * x_ix * ib_rails
    ib_res_eff = ix_2p5x
    ElseIf ib_width < 4 * x_ix Then
    ib_width_eff = 3 * x_ix * ib_rails
    ib_res_eff = ix_3x
    ElseIf ib_width < 5 * x_ix Then
    ib_width_eff = 4 * x_ix * ib_rails
    ib_res_eff = ix_4x
    Else
    ib_width_eff = (ib_width * ib_rails) / (5 * x_ix)
    ib_res_eff = ix_5x / ib_width_eff
    End If

Range("E50") = ia_res_eff

ma_width = Range("C51").Value
ma_rails = Range("D51").Value
    If ma_rails = 0 Then
    ma_res_eff = 1000000
    ElseIf ma_width < 1.5 * x_mx Then
    ma_width_eff = 1 * x_mx * ma_rails
    ma_res_eff = mx_1x
    ElseIf ma_width < 2 * x_mx Then
    ma_width_eff = 1.5 * x_mx * ma_rails
    ma_res_eff = mx_1p5x
    ElseIf ma_width < 2.5 * x_mx Then
    ma_width_eff = 2 * x_mx * ma_rails
    ma_res_eff = mx_2x
    ElseIf ma_width < 3 * x_mx Then
    ma_width_eff = 2.5 * x_mx * ma_rails
    ma_res_eff = mx_2p5x
    ElseIf ma_width < 4 * x_mx Then
    ma_width_eff = 3 * x_mx * ma_rails

    ma_res_eff = mx_3x

    ElseIf ma_width < 5 * x_mx Then

    ma_width_eff = 4 * x_mx * ma_rails

    ma_res_eff = mx_4x

    Else

    ma_width_eff = (ma_width * ma_rails) / (5 * x_mx)

    ma_res_eff = mx_5x / ma_width_eff

    End If


Range("E51") = ma_res_eff


mb_width = Range("C52").Value

mb_rails = Range("D52").Value

    If mb_rails = 0 Then

    mb_res_eff = 1000000

    ElseIf mb_width < 1.5 * x_mx Then

    mb_width_eff = 1 * x_mx * mb_rails

    mb_res_eff = mx_1x

    ElseIf mb_width < 2 * x_mx Then

    mb_width_eff = 1.5 * x_mx * mb_rails

    mb_res_eff = mx_1p5x

    ElseIf mb_width < 2.5 * x_mx Then

    mb_width_eff = 2 * x_mx * mb_rails

    mb_res_eff = mx_2x
    ElseIf mb_width < 3 * x_mx Then
    mb_width_eff = 2.5 * x_mx * mb_rails
    mb_res_eff = mx_2p5x
    ElseIf mb_width < 4 * x_mx Then
    mb_width_eff = 3 * x_mx * mb_rails
    mb_res_eff = mx_3x
    ElseIf mb_width < 5 * x_mx Then
    mb_width_eff = 4 * x_mx * mb_rails
    mb_res_eff = mx_4x
    Else
    mb_width_eff = (mb_width * mb_rails) / (5 * x_mx)
    mb_res_eff = mx_5x / mb_width_eff
    End If
    Range("E52") = mb_res_eff
hor_res_eff = 1 / ((1 / m2_res_eff) + (1 / m4_res_eff) + (1 / m6_res_eff) + (1 / m8_res_eff) + (1 / b2_res_eff) + (1 / ia_res_eff) + (1 / ma_res_eff))
vert_res_eff = 1 / ((1 / m1_res_eff) + (1 / m3_res_eff) + (1 / m5_res_eff) + (1 / m7_res_eff) + (1 / b1_res_eff) + (1 / ib_res_eff) + (1 / mb_res_eff))
numerator = ((2 * vert_res_eff) + hor_res_eff) * hor_res_eff
denominator = 2 * (vert_res_eff + hor_res_eff)
res_eff = numerator / denominator
Range("F38") = res_eff
End Sub

Sub diodeSupplyResCalc()
```

```
' vddResCalc Macro
' To calculate the resistance of vdd power bus

  Dim m1_1x, m1_1p5x, m1_2x, m1_2p5x, m1_3x, m1_4x, m1_5x, m1_iso As Double
  Dim mx_1x, mx_1p5x, mx_2x, mx_2p5x, mx_3x, mx_4x, mx_5x, mx_iso As Double
  Dim bx_1x, bx_1p5x, bx_2x, bx_2p5x, bx_3x, bx_4x, bx_5x, bx_iso As Double

  Dim e1_1x, e1_1p5x, e1_2x, e1_2p5x, e1_3x, e1_4x, e1_5x, e1_iso As Double
  Dim ix_1x, ix_1p5x, ix_2x, ix_2p5x, ix_3x, ix_4x, ix_5x, ix_iso As Double
  Dim ma_mb_1x, ma_mb_1p5x, ma_mb_2x, ma_mb_2p5x, ma_mb_3x, ma_mb_4x, ma_mb_5x, ma_mb_iso As Double

  Dim x_m1, x_mx, x_bx, x_e1, x_ix, x_ma_mb, width, width_eff, length As Double
  Dim m1_width_eff, m1_res_eff, mx_width_eff, mx_res_eff, bx_width_eff, bx_res_eff, e1_width_eff, e1_res_eff, ix_width_eff, ma_mb_width_eff As Double
  Dim metal As String
  Dim store As Range

Sheets("Placement_Calculator").Select

  'width = Range("B10").Value
  'metal = Range("B9").Value

Sheets("Technology_Inputs").Select
```

```
x_m1 = Range("C6").Value

x_mx = Range("C7").Value

x_bx = Range("C8").Value

x_e1 = Range("C9").Value

x_1x = Range("C10").Value

x_ma_mb = Range("C11").Value


m1_1x = Range("D6").Value

m1_1p5x = Range("E6").Value

m1_2x = Range("F6").Value

m1_2p5x = Range("G6").Value

m1_3x = Range("H6").Value

m1_4x = Range("I6").Value

m1_5x = Range("J6").Value

m1_iso = Range("K6").Value


mx_1x = Range("D7").Value

mx_1p5x = Range("E7").Value

mx_2x = Range("F7").Value

mx_2p5x = Range("G7").Value

mx_3x = Range("H7").Value

mx_4x = Range("I7").Value

mx_5x = Range("J7").Value

mx_iso = Range("K7").Value
```

```
bx_1x = Range("D8").Value
bx_1p5x = Range("E8").Value
bx_2x = Range("F8").Value
bx_2p5x = Range("G8").Value
bx_3x = Range("H8").Value
bx_4x = Range("I8").Value
bx_5x = Range("J8").Value
bx_iso = Range("K8").Value

e1_1x = Range("D9").Value
e1_1p5x = Range("E9").Value
e1_2x = Range("F9").Value
e1_2p5x = Range("G9").Value
e1_3x = Range("H9").Value
e1_4x = Range("I9").Value
e1_5x = Range("J9").Value
e1_iso = Range("K9").Value

ix_1x = Range("D10").Value
ix_1p5x = Range("E10").Value
ix_2x = Range("F10").Value
ix_2p5x = Range("G10").Value
ix_3x = Range("H10").Value
ix_4x = Range("I10").Value
ix_5x = Range("J10").Value
```

```
  ix_iso = Range("K10").Value

  ma_mb_1x = Range("D11").Value
  ma_mb_1p5x = Range("E11").Value
  ma_mb_2x = Range("F11").Value
  ma_mb_2p5x = Range("G11").Value
  ma_mb_3x = Range("H11").Value
  ma_mb_4x = Range("I11").Value
  ma_mb_5x = Range("J11").Value
  ma_mb_iso = Range("K11").Value

Sheets("Placement_Calculator").Select

m1_width = Range("C58").Value
m1_rails = Range("D58").Value
    If m1_rails = 0 Then
    m1_res_eff = 1000000
    ElseIf m1_width < 1.5 * x_m1 Then
    m1_width_eff = 1 * x_m1 * m1_rails
    m1_res_eff = m1_1x / m1_width_eff
    ElseIf m1_width < 2 * x_m1 Then
    m1_width_eff = 1.5 * x_m1 * m1_rails
    m1_res_eff = m1_1p5x / m1_width_eff
    ElseIf m1_width < 2.5 * x_m1 Then
    m1_width_eff = 2 * x_m1 * m1_rails
    m1_res_eff = m1_2x / m1_width_eff
    ElseIf m1_width < 3 * x_m1 Then
    m1_width_eff = 2.5 * x_m1 * m1_rails
    m1_res_eff = m1_2p5x / m1_width_eff
    ElseIf m1_width < 4 * x_m1 Then
    m1_width_eff = 3 * x_m1 * m1_rails
    m1_res_eff = m1_3x / m1_width_eff
    ElseIf m1_width < 5 * x_m1 Then
    m1_width_eff = 4 * x_m1 * m1_rails
    m1_res_eff = m1_4x / m1_width_eff
    Else
    m1_width_eff = (m1_width * m1_rails) / (5 * x_m1)
    m1_res_eff = m1_5x / m1_width_eff
    End If

Range("E58") = m1_res_eff


m2_width = Range("C59").Value
m2_rails = Range("D59").Value
    If m2_rails = 0 Then
    m2_res_eff = 1000000
    ElseIf m2_width < 1.5 * x_mx Then
    m2_width_eff = 1 * x_mx * m2_rails
    m2_res_eff = mx_1x / m2_width_eff
    ElseIf m2_width < 2 * x_mx Then
    m2_width_eff = 1.5 * x_mx * m2_rails

    m2_res_eff = mx_1p5x / m2_width_eff

    ElseIf m2_width < 2.5 * x_mx Then

    m2_width_eff = 2 * x_mx * m2_rails

    m2_res_eff = mx_2x / m2_width_eff

    ElseIf m2_width < 3 * x_mx Then

    m2_width_eff = 2.5 * x_mx * m2_rails

    m2_res_eff = mx_2p5x / m2_width_eff

    ElseIf m2_width < 4 * x_mx Then

    m2_width_eff = 3 * x_mx * m2_rails

    m2_res_eff = mx_3x / m2_width_eff

    ElseIf m2_width < 5 * x_mx Then

    m2_width_eff = 4 * x_mx * m2_rails

    m2_res_eff = mx_4x / m2_width_eff

    Else

    m2_width_eff = (m2_width * m2_rails) / (5 * x_mx)

    m2_res_eff = mx_5x / m2_width_eff

    End If


Range("E59") = m2_res_eff


m3_width = Range("C60").Value

m3_rails = Range("D60").Value

    If m3_rails = 0 Then

    m3_res_eff = 1000000
    ElseIf m3_width < 1.5 * x_mx Then
    m3_width_eff = 1 * x_mx * m3_rails
    m3_res_eff = mx_1x / m3_width_eff
    ElseIf m3_width < 2 * x_mx Then
    m3_width_eff = 1.5 * x_mx * m3_rails
    m3_res_eff = mx_1p5x / m3_width_eff
    ElseIf m3_width < 2.5 * x_mx Then
    m3_width_eff = 2 * x_mx * m3_rails
    m3_res_eff = mx_2x / m3_width_eff
    ElseIf m3_width < 3 * x_mx Then
    m3_width_eff = 2.5 * x_mx * m3_rails
    m3_res_eff = mx_2p5x / m3_width_eff
    ElseIf m3_width < 4 * x_mx Then
    m3_width_eff = 3 * x_mx * m3_rails
    m3_res_eff = mx_3x / m3_width_eff
    ElseIf m3_width < 5 * x_mx Then
    m3_width_eff = 4 * x_mx * m3_rails
    m3_res_eff = mx_4x / m3_width_eff
    Else
    m3_width_eff = (m3_width * m3_rails) / (5 * x_mx)
    m3_res_eff = mx_5x / m3_width_eff
    End If

Range("E60") = m3_res_eff
```

```
m4_width = Range("C61").Value

m4_rails = Range("D61").Value

    If m4_rails = 0 Then

    m4_res_eff = 1000000

    ElseIf m4_width < 1.5 * x_mx Then

    m4_width_eff = 1 * x_mx * m4_rails

    m4_res_eff = mx_1x / m4_width_eff

    ElseIf m4_width < 2 * x_mx Then

    m4_width_eff = 1.5 * x_mx * m4_rails

    m4_res_eff = mx_1p5x / m4_width_eff

    ElseIf m4_width < 2.5 * x_mx Then

    m4_width_eff = 2 * x_mx * m4_rails

    m4_res_eff = mx_2x / m4_width_eff

    ElseIf m4_width < 3 * x_mx Then

    m4_width_eff = 2.5 * x_mx * m4_rails

    m4_res_eff = mx_2p5x / m4_width_eff

    ElseIf m4_width < 4 * x_mx Then

    m4_width_eff = 3 * x_mx * m4_rails

    m4_res_eff = mx_3x / m4_width_eff

    ElseIf m4_width < 5 * x_mx Then

    m4_width_eff = 4 * x_mx * m4_rails

    m4_res_eff = mx_4x / m4_width_eff

    Else

    m4_width_eff = (m4_width * m4_rails) / (5 * x_mx)

    m4_res_eff = mx_5x / m4_width_eff
    End If


Range("E61") = m4_res_eff


m5_width = Range("C62").Value
m5_rails = Range("D62").Value
    If m5_rails = 0 Then
    m5_res_eff = 1000000
    ElseIf m5_width < 1.5 * x_mx Then
    m5_width_eff = 1 * x_mx * m5_rails
    m5_res_eff = mx_1x / m5_width_eff
    ElseIf m5_width < 2 * x_mx Then
    m5_width_eff = 1.5 * x_mx * m5_rails
    m5_res_eff = mx_1p5x / m5_width_eff
    ElseIf m5_width < 2.5 * x_mx Then
    m5_width_eff = 2 * x_mx * m5_rails
    m5_res_eff = mx_2x / m5_width_eff
    ElseIf m5_width < 3 * x_mx Then
    m5_width_eff = 2.5 * x_mx * m5_rails
    m5_res_eff = mx_2p5x / m5_width_eff
    ElseIf m5_width < 4 * x_mx Then
    m5_width_eff = 3 * x_mx * m5_rails
    m5_res_eff = mx_3x / m5_width_eff
    ElseIf m5_width < 5 * x_mx Then
    m5_width_eff = 4 * x_mx * m5_rails
    m5_res_eff = mx_4x / m5_width_eff

    Else

    m5_width_eff = (m5_width * m5_rails) / (5 * x_mx)

    m5_res_eff = mx_5x / m5_width_eff

    End If


Range("E62") = m5_res_eff


m6_width = Range("C63").Value

m6_rails = Range("D63").Value

    If m6_rails = 0 Then

    m6_res_eff = 1000000

    ElseIf m6_width < 1.5 * x_mx Then

    m6_width_eff = 1 * x_mx * m6_rails

    m6_res_eff = mx_1x / m6_width_eff

    ElseIf m6_width < 2 * x_mx Then

    m6_width_eff = 1.5 * x_mx * m6_rails

    m6_res_eff = mx_1p5x / m6_width_eff

    ElseIf m6_width < 2.5 * x_mx Then

    m6_width_eff = 2 * x_mx * m6_rails

    m6_res_eff = mx_2x / m6_width_eff

    ElseIf m6_width < 3 * x_mx Then

    m6_width_eff = 2.5 * x_mx * m6_rails

    m6_res_eff = mx_2p5x / m6_width_eff

    ElseIf m6_width < 4 * x_mx Then
    m6_width_eff = 3 * x_mx * m6_rails
    m6_res_eff = mx_3x / m6_width_eff
    ElseIf m6_width < 5 * x_mx Then
    m6_width_eff = 4 * x_mx * m6_rails
    m6_res_eff = mx_4x / m6_width_eff
    Else
    m6_width_eff = (m6_width * m6_rails) / (5 * x_mx)
    m6_res_eff = mx_5x / m6_width_eff
    End If

Range("E63") = m6_res_eff

m7_width = Range("C64").Value
m7_rails = Range("D64").Value
    If m7_rails = 0 Then
    m7_res_eff = 1000000
    ElseIf m7_width < 1.5 * x_mx Then
    m7_width_eff = 1 * x_mx * m7_rails
    m7_res_eff = mx_1x / m7_width_eff
    ElseIf m7_width < 2 * x_mx Then
    m7_width_eff = 1.5 * x_mx * m7_rails
    m7_res_eff = mx_1p5x / m7_width_eff
    ElseIf m7_width < 2.5 * x_mx Then
    m7_width_eff = 2 * x_mx * m7_rails
    m7_res_eff = mx_2x / m7_width_eff
    ElseIf m7_width < 3 * x_mx Then

    m7_width_eff = 2.5 * x_mx * m7_rails

    m7_res_eff = mx_2p5x / m7_width_eff

    ElseIf m7_width < 4 * x_mx Then

    m7_width_eff = 3 * x_mx * m7_rails

    m7_res_eff = mx_3x / m7_width_eff

    ElseIf m7_width < 5 * x_mx Then

    m7_width_eff = 4 * x_mx * m7_rails

    m7_res_eff = mx_4x / m7_width_eff

    Else

    m7_width_eff = (m7_width * m7_rails) / (5 * x_mx)

    m7_res_eff = mx_5x / m7_width_eff

    End If


Range("E64") = m7_res_eff


m8_width = Range("C65").Value

m8_rails = Range("D65").Value

    If m8_rails = 0 Then

    m8_res_eff = 1000000

    ElseIf m7_width < 1.5 * x_mx Then

    m8_width_eff = 1 * x_mx * m8_rails

    m8_res_eff = mx_1x / m8_width_eff

    ElseIf m7_width < 2 * x_mx Then

    m8_width_eff = 1.5 * x_mx * m8_rails
    m8_res_eff = mx_1p5x / m8_width_eff

    ElseIf m7_width < 2.5 * x_mx Then

    m8_width_eff = 2 * x_mx * m8_rails

    m8_res_eff = mx_2x / m8_width_eff

    ElseIf m7_width < 3 * x_mx Then

    m8_width_eff = 2.5 * x_mx * m8_rails

    m8_res_eff = mx_2p5x / m8_width_eff

    ElseIf m7_width < 4 * x_mx Then

    m8_width_eff = 3 * x_mx * m8_rails

    m8_res_eff = mx_3x / m8_width_eff

    ElseIf m7_width < 5 * x_mx Then

    m8_width_eff = 4 * x_mx * m8_rails

    m8_res_eff = mx_4x / m8_width_eff

    Else

    m8_width_eff = (m8_width * m8_rails) / (5 * x_mx)

    m8_res_eff = mx_5x / m8_width_eff

    End If


Range("E65") = m8_res_eff


b1_width = Range("C66").Value

b1_rails = Range("D66").Value

    If b1_rails = 0 Then

    b1_res_eff = 1000000

    ElseIf b1_width < 1.5 * x_bx Then
    b1_width_eff = 1 * x_bx * b1_rails
    b1_res_eff = bx_1x
    ElseIf b1_width < 2 * x_bx Then
    b1_width_eff = 1.5 * x_bx * b1_rails
    b1_res_eff = bx_1p5x
    ElseIf b1_width < 2.5 * x_bx Then
    b1_width_eff = 2 * x_bx * b1_rails
    b1_res_eff = bx_2x
    ElseIf b1_width < 3 * x_bx Then
    b1_width_eff = 2.5 * x_bx * b1_rails
    b1_res_eff = bx_2p5x
    ElseIf b1_width < 4 * x_bx Then
    b1_width_eff = 3 * x_bx * b1_rails
    b1_res_eff = bx_3x
    ElseIf b1_width < 5 * x_bx Then
    b1_width_eff = 4 * x_bx * b1_rails
    b1_res_eff = bx_4x
    Else
    b1_width_eff = (b1_width * b1_rails) / (5 * x_bx)
    b1_res_eff = bx_5x / b1_width_eff
    End If

Range("E66") = b1_res_eff

b2_width = Range("C67").Value
```

```
b2_rails = Range("D67").Value
    If b2_rails = 0 Then
    b2_res_eff = 1000000
    ElseIf b2_width < 1.5 * x_bx Then
    b2_width_eff = 1 * x_bx * b2_rails
    b2_res_eff = bx_1x
    ElseIf b2_width < 2 * x_bx Then
    b2_width_eff = 1.5 * x_bx * b2_rails
    b2_res_eff = bx_1p5x
    ElseIf b2_width < 2.5 * x_bx Then
    b2_width_eff = 2 * x_bx * b2_rails
    b2_res_eff = bx_2x
    ElseIf b2_width < 3 * x_bx Then
    b2_width_eff = 2.5 * x_bx * b2_rails
    b2_res_eff = bx_2p5x
    ElseIf b2_width < 4 * x_bx Then
    b2_width_eff = 3 * x_bx * b2_rails
    b2_res_eff = bx_3x
    ElseIf b2_width < 5 * x_bx Then
    b2_width_eff = 4 * x_bx * b2_rails
    b2_res_eff = bx_4x
    Else
    b2_width_eff = (b2_width * b2_rails) / (5 * x_bx)
    b2_res_eff = bx_5x / b2_width_eff
    End If
```

```
Range("E67") = b2_res_eff


e1_width = Range("C68").Value

e1_rails = Range("D68").Value

  If e1_rails = 0 Then

  e1_res_eff = 1000000

  ElseIf e1_width < 1.5 * x_e1 Then

  e1_width_eff = 1 * x_e1 * e1_rails

  e1_res_eff = e1_1x

  ElseIf e1_width < 2 * x_e1 Then

  e1_width_eff = 1.5 * x_e1 * e1_rails

  e1_res_eff = e1_1p5x

  ElseIf e1_width < 2.5 * x_e1 Then

  e1_width_eff = 2 * x_e1 * e1_rails

  e1_res_eff = e1_2x

  ElseIf e1_width < 3 * x_e1 Then

  e1_width_eff = 2.5 * x_e1 * e1_rails

  e1_res_eff = e1_2p5x

  ElseIf e1_width < 4 * x_e1 Then

  e1_width_eff = 3 * x_e1 * e1_rails

  e1_res_eff = e1_3x

  ElseIf e1_width < 5 * x_e1 Then

  e1_width_eff = 4 * x_e1 * e1_rails

  e1_res_eff = e1_4x
```

```
    Else
    e1_width_eff = (e1_width * e1_rails) / (5 * x_e1)
    e1_res_eff = e1_5x / e1_width_eff
    End If

Range("E68") = e1_res_eff


ia_width = Range("C69").Value
ia_rails = Range("D69").Value
    If ia_rails = 0 Then
    ia_res_eff = 1000000
    ElseIf ia_width < 1.5 * x_ix Then
    ia_width_eff = 1 * x_ix * ia_rails
    ia_res_eff = ix_1x
    ElseIf ia_width < 2 * x_ix Then
    ia_width_eff = 1.5 * x_ix * ia_rails
    ia_res_eff = ix_1p5x
    ElseIf ia_width < 2.5 * x_ix Then
    ia_width_eff = 2 * x_ix * ia_rails
    ia_res_eff = ix_2x
    ElseIf ia_width < 3 * x_ix Then
    ia_width_eff = 2.5 * x_ix * ia_rails
    ia_res_eff = ix_2p5x
    ElseIf ia_width < 4 * x_ix Then
    ia_width_eff = 3 * x_ix * ia_rails
    ia_res_eff = ix_3x

    ElseIf ia_width < 5 * x_ix Then

    ia_width_eff = 4 * x_ix * ia_rails

    ia_res_eff = ix_4x

    Else

    ia_width_eff = (ia_width * ia_rails) / (5 * x_ix)

    ia_res_eff = ix_5x / ia_width_eff

    End If


Range("E69") = ia_res_eff


ib_width = Range("C70").Value

ib_rails = Range("D70").Value

    If ib_rails = 0 Then

    ib_res_eff = 1000000

    ElseIf ib_width < 1.5 * x_ix Then

    ib_width_eff = 1 * x_ix * ib_rails

    ib_res_eff = ix_1x

    ElseIf ib_width < 2 * x_ix Then

    ib_width_eff = 1.5 * x_ix * ib_rails

    ib_res_eff = ix_1p5x

    ElseIf ib_width < 2.5 * x_ix Then

    ib_width_eff = 2 * x_ix * ib_rails

    ib_res_eff = ix_2x

    ElseIf ib_width < 3 * x_ix Then
    ib_width_eff = 2.5 * x_ix * ib_rails

    ib_res_eff = ix_2p5x

    ElseIf ib_width < 4 * x_ix Then

    ib_width_eff = 3 * x_ix * ib_rails

    ib_res_eff = ix_3x

    ElseIf ib_width < 5 * x_ix Then

    ib_width_eff = 4 * x_ix * ib_rails

    ib_res_eff = ix_4x

    Else

    ib_width_eff = (ib_width * ib_rails) / (5 * x_ix)

    ib_res_eff = ix_5x / ib_width_eff

    End If


Range("E70") = ia_res_eff


ma_width = Range("C71").Value

ma_rails = Range("D71").Value

    If ma_rails = 0 Then

    ma_res_eff = 1000000

    ElseIf ma_width < 1.5 * x_mx Then

    ma_width_eff = 1 * x_mx * ma_rails

    ma_res_eff = mx_1x

    ElseIf ma_width < 2 * x_mx Then

    ma_width_eff = 1.5 * x_mx * ma_rails

    ma_res_eff = mx_1p5x
    ElseIf ma_width < 2.5 * x_mx Then
    ma_width_eff = 2 * x_mx * ma_rails
    ma_res_eff = mx_2x
    ElseIf ma_width < 3 * x_mx Then
    ma_width_eff = 2.5 * x_mx * ma_rails
    ma_res_eff = mx_2p5x
    ElseIf ma_width < 4 * x_mx Then
    ma_width_eff = 3 * x_mx * ma_rails
    ma_res_eff = mx_3x
    ElseIf ma_width < 5 * x_mx Then
    ma_width_eff = 4 * x_mx * ma_rails
    ma_res_eff = mx_4x
    Else
    ma_width_eff = (ma_width * ma_rails) / (5 * x_mx)
    ma_res_eff = mx_5x / ma_width_eff
    End If

Range("E71") = ma_res_eff


mb_width = Range("C72").Value
mb_rails = Range("D72").Value
    If mb_rails = 0 Then
    mb_res_eff = 1000000
    ElseIf mb_width < 1.5 * x_mx Then
    mb_width_eff = 1 * x_mx * mb_rails
    mb_res_eff = mx_1x
    ElseIf mb_width < 2 * x_mx Then
    mb_width_eff = 1.5 * x_mx * mb_rails
    mb_res_eff = mx_1p5x
    ElseIf mb_width < 2.5 * x_mx Then
    mb_width_eff = 2 * x_mx * mb_rails
    mb_res_eff = mx_2x
    ElseIf mb_width < 3 * x_mx Then
    mb_width_eff = 2.5 * x_mx * mb_rails
    mb_res_eff = mx_2p5x
    ElseIf mb_width < 4 * x_mx Then
    mb_width_eff = 3 * x_mx * mb_rails
    mb_res_eff = mx_3x
    ElseIf mb_width < 5 * x_mx Then
    mb_width_eff = 4 * x_mx * mb_rails
    mb_res_eff = mx_4x
    Else
    mb_width_eff = (mb_width * mb_rails) / (5 * x_mx)
    mb_res_eff = mx_5x / mb_width_eff
    End If

Range("E72") = mb_res_eff

hor_res_eff = 1 / ((1 / m2_res_eff) + (1 / m4_res_eff) + (1 / m6_res_eff) + (1 / m8_res_eff) + (1 / b2_res_eff) + (1 / ia_res_eff) + (1 / ma_res_eff))
```

```
vert_res_eff = 1 / ((1 / m1_res_eff) + (1 / m3_res_eff) + (1 / m5_res_eff) + (1 / m7_res_eff) + (1 / b1_res_eff) + (1 / ib_res_eff) + (1 / mb_res_eff))
numerator = ((2 * vert_res_eff) + hor_res_eff) * hor_res_eff
denominator = 2 * (vert_res_eff + hor_res_eff)
res_eff = numerator / denominator

Range("F58") = res_eff

End Sub

Sub padDiodeResCalc()
' vddResCalc Macro
' To calculate the resistance of vdd power bus

  Dim m1_1x, m1_1p5x, m1_2x, m1_2p5x, m1_3x, m1_4x, m1_5x, m1_iso As Double
  Dim mx_1x, mx_1p5x, mx_2x, mx_2p5x, mx_3x, mx_4x, mx_5x, mx_iso As Double
  Dim bx_1x, bx_1p5x, bx_2x, bx_2p5x, bx_3x, bx_4x, bx_5x, bx_iso As Double

  Dim e1_1x, e1_1p5x, e1_2x, e1_2p5x, e1_3x, e1_4x, e1_5x, e1_iso As Double
  Dim ix_1x, ix_1p5x, ix_2x, ix_2p5x, ix_3x, ix_4x, ix_5x, ix_iso As Double
  Dim ma_mb_1x, ma_mb_1p5x, ma_mb_2x, ma_mb_2p5x, ma_mb_3x, ma_mb_4x, ma_mb_5x, ma_mb_iso As Double

  Dim x_m1, x_mx, x_bx, x_e1, x_ix, x_ma_mb, width, width_eff, length As Double
  Dim m1_width_eff, m1_res_eff, mx_width_eff, mx_res_eff, bx_width_eff, bx_res_eff, e1_width_eff, e1_res_eff, ix_width_eff, ma_mb_width_eff As Double
  Dim metal As String
  Dim store As Range



Sheets("Placement_Calculator").Select

  'width = Range("B10").Value
  'metal = Range("B9").Value

Sheets("Technology_Inputs").Select

  x_m1 = Range("C6").Value
  x_mx = Range("C7").Value
  x_bx = Range("C8").Value
  x_e1 = Range("C9").Value
  x_1x = Range("C10").Value
  x_ma_mb = Range("C11").Value

  m1_1x = Range("D6").Value
  m1_1p5x = Range("E6").Value
  m1_2x = Range("F6").Value
  m1_2p5x = Range("G6").Value
  m1_3x = Range("H6").Value
```

```
m1_4x = Range("I6").Value
m1_5x = Range("J6").Value
m1_iso = Range("K6").Value

mx_1x = Range("D7").Value
mx_1p5x = Range("E7").Value
mx_2x = Range("F7").Value
mx_2p5x = Range("G7").Value
mx_3x = Range("H7").Value
mx_4x = Range("I7").Value
mx_5x = Range("J7").Value
mx_iso = Range("K7").Value

bx_1x = Range("D8").Value
bx_1p5x = Range("E8").Value
bx_2x = Range("F8").Value
bx_2p5x = Range("G8").Value
bx_3x = Range("H8").Value
bx_4x = Range("I8").Value
bx_5x = Range("J8").Value
bx_iso = Range("K8").Value

e1_1x = Range("D9").Value
e1_1p5x = Range("E9").Value
e1_2x = Range("F9").Value
```

```
 e1_2p5x = Range("G9").Value

 e1_3x = Range("H9").Value

 e1_4x = Range("I9").Value

 e1_5x = Range("J9").Value

 e1_iso = Range("K9").Value


 ix_1x = Range("D10").Value

 ix_1p5x = Range("E10").Value

 ix_2x = Range("F10").Value

 ix_2p5x = Range("G10").Value

 ix_3x = Range("H10").Value

 ix_4x = Range("I10").Value

 ix_5x = Range("J10").Value

 ix_iso = Range("K10").Value


 ma_mb_1x = Range("D11").Value

 ma_mb_1p5x = Range("E11").Value

 ma_mb_2x = Range("F11").Value

 ma_mb_2p5x = Range("G11").Value

 ma_mb_3x = Range("H11").Value

 ma_mb_4x = Range("I11").Value

 ma_mb_5x = Range("J11").Value

 ma_mb_iso = Range("K11").Value


Sheets("Placement_Calculator").Select
```

```
m1_width = Range("C78").Value
m1_rails = Range("D78").Value
    If m1_rails = 0 Then
    m1_res_eff = 1000000
    ElseIf m1_width < 1.5 * x_m1 Then
    m1_width_eff = 1 * x_m1 * m1_rails
    m1_res_eff = m1_1x / m1_width_eff
    ElseIf m1_width < 2 * x_m1 Then
    m1_width_eff = 1.5 * x_m1 * m1_rails
    m1_res_eff = m1_1p5x / m1_width_eff
    ElseIf m1_width < 2.5 * x_m1 Then
    m1_width_eff = 2 * x_m1 * m1_rails
    m1_res_eff = m1_2x / m1_width_eff
    ElseIf m1_width < 3 * x_m1 Then
    m1_width_eff = 2.5 * x_m1 * m1_rails
    m1_res_eff = m1_2p5x / m1_width_eff
    ElseIf m1_width < 4 * x_m1 Then
    m1_width_eff = 3 * x_m1 * m1_rails
    m1_res_eff = m1_3x / m1_width_eff
    ElseIf m1_width < 5 * x_m1 Then
    m1_width_eff = 4 * x_m1 * m1_rails
    m1_res_eff = m1_4x / m1_width_eff
    Else
    m1_width_eff = (m1_width * m1_rails) / (5 * x_m1)
    m1_res_eff = m1_5x / m1_width_eff
    End If

Range("E78") = m1_res_eff

m2_width = Range("C79").Value
m2_rails = Range("D79").Value
    If m2_rails = 0 Then
    m2_res_eff = 1000000
    ElseIf m2_width < 1.5 * x_mx Then
    m2_width_eff = 1 * x_mx * m2_rails
    m2_res_eff = mx_1x / m2_width_eff
    ElseIf m2_width < 2 * x_mx Then
    m2_width_eff = 1.5 * x_mx * m2_rails
    m2_res_eff = mx_1p5x / m2_width_eff
    ElseIf m2_width < 2.5 * x_mx Then
    m2_width_eff = 2 * x_mx * m2_rails
    m2_res_eff = mx_2x / m2_width_eff
    ElseIf m2_width < 3 * x_mx Then
    m2_width_eff = 2.5 * x_mx * m2_rails
    m2_res_eff = mx_2p5x / m2_width_eff
    ElseIf m2_width < 4 * x_mx Then
    m2_width_eff = 3 * x_mx * m2_rails
    m2_res_eff = mx_3x / m2_width_eff
    ElseIf m2_width < 5 * x_mx Then
    m2_width_eff = 4 * x_mx * m2_rails
    m2_res_eff = mx_4x / m2_width_eff
    Else
    m2_width_eff = (m2_width * m2_rails) / (5 * x_mx)
    m2_res_eff = mx_5x / m2_width_eff
    End If

Range("E79") = m2_res_eff

m3_width = Range("C80").Value
m3_rails = Range("D80").Value
    If m3_rails = 0 Then
    m3_res_eff = 1000000
    ElseIf m3_width < 1.5 * x_mx Then
    m3_width_eff = 1 * x_mx * m3_rails
    m3_res_eff = mx_1x / m3_width_eff
    ElseIf m3_width < 2 * x_mx Then
    m3_width_eff = 1.5 * x_mx * m3_rails
    m3_res_eff = mx_1p5x / m3_width_eff
    ElseIf m3_width < 2.5 * x_mx Then
    m3_width_eff = 2 * x_mx * m3_rails
    m3_res_eff = mx_2x / m3_width_eff
    ElseIf m3_width < 3 * x_mx Then
    m3_width_eff = 2.5 * x_mx * m3_rails
    m3_res_eff = mx_2p5x / m3_width_eff
    ElseIf m3_width < 4 * x_mx Then
    m3_width_eff = 3 * x_mx * m3_rails
    m3_res_eff = mx_3x / m3_width_eff
    ElseIf m3_width < 5 * x_mx Then
    m3_width_eff = 4 * x_mx * m3_rails
    m3_res_eff = mx_4x / m3_width_eff
    Else
    m3_width_eff = (m3_width * m3_rails) / (5 * x_mx)
    m3_res_eff = mx_5x / m3_width_eff
    End If

Range("E80") = m3_res_eff


m4_width = Range("C81").Value
m4_rails = Range("D81").Value
    If m4_rails = 0 Then
    m4_res_eff = 1000000
    ElseIf m4_width < 1.5 * x_mx Then
    m4_width_eff = 1 * x_mx * m4_rails
    m4_res_eff = mx_1x / m4_width_eff
    ElseIf m4_width < 2 * x_mx Then
    m4_width_eff = 1.5 * x_mx * m4_rails
    m4_res_eff = mx_1p5x / m4_width_eff
    ElseIf m4_width < 2.5 * x_mx Then
    m4_width_eff = 2 * x_mx * m4_rails
    m4_res_eff = mx_2x / m4_width_eff

    ElseIf m4_width < 3 * x_mx Then

    m4_width_eff = 2.5 * x_mx * m4_rails

    m4_res_eff = mx_2p5x / m4_width_eff

    ElseIf m4_width < 4 * x_mx Then

    m4_width_eff = 3 * x_mx * m4_rails

    m4_res_eff = mx_3x / m4_width_eff

    ElseIf m4_width < 5 * x_mx Then

    m4_width_eff = 4 * x_mx * m4_rails

    m4_res_eff = mx_4x / m4_width_eff

    Else

    m4_width_eff = (m4_width * m4_rails) / (5 * x_mx)

    m4_res_eff = mx_5x / m4_width_eff

    End If


Range("E81") = m4_res_eff


m5_width = Range("C82").Value

m5_rails = Range("D82").Value

    If m5_rails = 0 Then

    m5_res_eff = 1000000

    ElseIf m5_width < 1.5 * x_mx Then

    m5_width_eff = 1 * x_mx * m5_rails

    m5_res_eff = mx_1x / m5_width_eff

    ElseIf m5_width < 2 * x_mx Then
    m5_width_eff = 1.5 * x_mx * m5_rails

    m5_res_eff = mx_1p5x / m5_width_eff

    ElseIf m5_width < 2.5 * x_mx Then

    m5_width_eff = 2 * x_mx * m5_rails

    m5_res_eff = mx_2x / m5_width_eff

    ElseIf m5_width < 3 * x_mx Then

    m5_width_eff = 2.5 * x_mx * m5_rails

    m5_res_eff = mx_2p5x / m5_width_eff

    ElseIf m5_width < 4 * x_mx Then

    m5_width_eff = 3 * x_mx * m5_rails

    m5_res_eff = mx_3x / m5_width_eff

    ElseIf m5_width < 5 * x_mx Then

    m5_width_eff = 4 * x_mx * m5_rails

    m5_res_eff = mx_4x / m5_width_eff

    Else

    m5_width_eff = (m5_width * m5_rails) / (5 * x_mx)

    m5_res_eff = mx_5x / m5_width_eff

    End If


Range("E82") = m5_res_eff


m6_width = Range("C83").Value

m6_rails = Range("D83").Value

    If m6_rails = 0 Then

    m6_res_eff = 1000000
    ElseIf m6_width < 1.5 * x_mx Then
    m6_width_eff = 1 * x_mx * m6_rails
    m6_res_eff = mx_1x / m6_width_eff
    ElseIf m6_width < 2 * x_mx Then
    m6_width_eff = 1.5 * x_mx * m6_rails
    m6_res_eff = mx_1p5x / m6_width_eff
    ElseIf m6_width < 2.5 * x_mx Then
    m6_width_eff = 2 * x_mx * m6_rails
    m6_res_eff = mx_2x / m6_width_eff
    ElseIf m6_width < 3 * x_mx Then
    m6_width_eff = 2.5 * x_mx * m6_rails
    m6_res_eff = mx_2p5x / m6_width_eff
    ElseIf m6_width < 4 * x_mx Then
    m6_width_eff = 3 * x_mx * m6_rails
    m6_res_eff = mx_3x / m6_width_eff
    ElseIf m6_width < 5 * x_mx Then
    m6_width_eff = 4 * x_mx * m6_rails
    m6_res_eff = mx_4x / m6_width_eff
    Else
    m6_width_eff = (m6_width * m6_rails) / (5 * x_mx)
    m6_res_eff = mx_5x / m6_width_eff
    End If

Range("E83") = m6_res_eff
```

```
m7_width = Range("C84").Value
m7_rails = Range("D84").Value
    If m7_rails = 0 Then
    m7_res_eff = 1000000
    ElseIf m7_width < 1.5 * x_mx Then
    m7_width_eff = 1 * x_mx * m7_rails
    m7_res_eff = mx_1x / m7_width_eff
    ElseIf m7_width < 2 * x_mx Then
    m7_width_eff = 1.5 * x_mx * m7_rails
    m7_res_eff = mx_1p5x / m7_width_eff
    ElseIf m7_width < 2.5 * x_mx Then
    m7_width_eff = 2 * x_mx * m7_rails
    m7_res_eff = mx_2x / m7_width_eff
    ElseIf m7_width < 3 * x_mx Then
    m7_width_eff = 2.5 * x_mx * m7_rails
    m7_res_eff = mx_2p5x / m7_width_eff
    ElseIf m7_width < 4 * x_mx Then
    m7_width_eff = 3 * x_mx * m7_rails
    m7_res_eff = mx_3x / m7_width_eff
    ElseIf m7_width < 5 * x_mx Then
    m7_width_eff = 4 * x_mx * m7_rails
    m7_res_eff = mx_4x / m7_width_eff
    Else
    m7_width_eff = (m7_width * m7_rails) / (5 * x_mx)
    m7_res_eff = mx_5x / m7_width_eff
    End If

Range("E84") = m7_res_eff

m8_width = Range("C85").Value
m8_rails = Range("D85").Value
    If m8_rails = 0 Then
    m8_res_eff = 1000000
    ElseIf m7_width < 1.5 * x_mx Then
    m8_width_eff = 1 * x_mx * m8_rails
    m8_res_eff = mx_1x / m8_width_eff
    ElseIf m7_width < 2 * x_mx Then
    m8_width_eff = 1.5 * x_mx * m8_rails
    m8_res_eff = mx_1p5x / m8_width_eff
    ElseIf m7_width < 2.5 * x_mx Then
    m8_width_eff = 2 * x_mx * m8_rails
    m8_res_eff = mx_2x / m8_width_eff
    ElseIf m7_width < 3 * x_mx Then
    m8_width_eff = 2.5 * x_mx * m8_rails
    m8_res_eff = mx_2p5x / m8_width_eff
    ElseIf m7_width < 4 * x_mx Then
    m8_width_eff = 3 * x_mx * m8_rails
    m8_res_eff = mx_3x / m8_width_eff
    ElseIf m7_width < 5 * x_mx Then
    m8_width_eff = 4 * x_mx * m8_rails
    m8_res_eff = mx_4x / m8_width_eff
    Else
    m8_width_eff = (m8_width * m8_rails) / (5 * x_mx)
    m8_res_eff = mx_5x / m8_width_eff
    End If

Range("E85") = m8_res_eff

b1_width = Range("C86").Value
b1_rails = Range("D86").Value
    If b1_rails = 0 Then
    b1_res_eff = 1000000
    ElseIf b1_width < 1.5 * x_bx Then
    b1_width_eff = 1 * x_bx * b1_rails
    b1_res_eff = bx_1x
    ElseIf b1_width < 2 * x_bx Then
    b1_width_eff = 1.5 * x_bx * b1_rails
    b1_res_eff = bx_1p5x
    ElseIf b1_width < 2.5 * x_bx Then
    b1_width_eff = 2 * x_bx * b1_rails
    b1_res_eff = bx_2x
    ElseIf b1_width < 3 * x_bx Then
    b1_width_eff = 2.5 * x_bx * b1_rails
    b1_res_eff = bx_2p5x
    ElseIf b1_width < 4 * x_bx Then
    b1_width_eff = 3 * x_bx * b1_rails
    b1_res_eff = bx_3x
    ElseIf b1_width < 5 * x_bx Then
    b1_width_eff = 4 * x_bx * b1_rails
    b1_res_eff = bx_4x
    Else
    b1_width_eff = (b1_width * b1_rails) / (5 * x_bx)
    b1_res_eff = bx_5x / b1_width_eff
    End If

Range("E86") = b1_res_eff

b2_width = Range("C87").Value
b2_rails = Range("D87").Value
    If b2_rails = 0 Then
    b2_res_eff = 1000000
    ElseIf b2_width < 1.5 * x_bx Then
    b2_width_eff = 1 * x_bx * b2_rails
    b2_res_eff = bx_1x
    ElseIf b2_width < 2 * x_bx Then
    b2_width_eff = 1.5 * x_bx * b2_rails
    b2_res_eff = bx_1p5x
    ElseIf b2_width < 2.5 * x_bx Then
    b2_width_eff = 2 * x_bx * b2_rails
    b2_res_eff = bx_2x
    ElseIf b2_width < 3 * x_bx Then
    b2_width_eff = 2.5 * x_bx * b2_rails
    b2_res_eff = bx_2p5x
    ElseIf b2_width < 4 * x_bx Then
    b2_width_eff = 3 * x_bx * b2_rails
    b2_res_eff = bx_3x
    ElseIf b2_width < 5 * x_bx Then
    b2_width_eff = 4 * x_bx * b2_rails
    b2_res_eff = bx_4x
    Else
    b2_width_eff = (b2_width * b2_rails) / (5 * x_bx)
    b2_res_eff = bx_5x / b2_width_eff
    End If

Range("E87") = b2_res_eff


e1_width = Range("C88").Value
e1_rails = Range("D88").Value
    If e1_rails = 0 Then
    e1_res_eff = 1000000
    ElseIf e1_width < 1.5 * x_e1 Then
    e1_width_eff = 1 * x_e1 * e1_rails
    e1_res_eff = e1_1x
    ElseIf e1_width < 2 * x_e1 Then
    e1_width_eff = 1.5 * x_e1 * e1_rails
    e1_res_eff = e1_1p5x
    ElseIf e1_width < 2.5 * x_e1 Then
    e1_width_eff = 2 * x_e1 * e1_rails
    e1_res_eff = e1_2x
    ElseIf e1_width < 3 * x_e1 Then
    e1_width_eff = 2.5 * x_e1 * e1_rails
    e1_res_eff = e1_2p5x
    ElseIf e1_width < 4 * x_e1 Then
    e1_width_eff = 3 * x_e1 * e1_rails
    e1_res_eff = e1_3x
    ElseIf e1_width < 5 * x_e1 Then
    e1_width_eff = 4 * x_e1 * e1_rails
    e1_res_eff = e1_4x
    Else
    e1_width_eff = (e1_width * e1_rails) / (5 * x_e1)
    e1_res_eff = e1_5x / e1_width_eff
    End If

Range("E88") = e1_res_eff


ia_width = Range("C89").Value
ia_rails = Range("D89").Value
    If ia_rails = 0 Then
    ia_res_eff = 1000000
    ElseIf ia_width < 1.5 * x_ix Then
    ia_width_eff = 1 * x_ix * ia_rails

    ia_res_eff = ix_1x

    ElseIf ia_width < 2 * x_ix Then

    ia_width_eff = 1.5 * x_ix * ia_rails

    ia_res_eff = ix_1p5x

    ElseIf ia_width < 2.5 * x_ix Then

    ia_width_eff = 2 * x_ix * ia_rails

    ia_res_eff = ix_2x

    ElseIf ia_width < 3 * x_ix Then

    ia_width_eff = 2.5 * x_ix * ia_rails

    ia_res_eff = ix_2p5x

    ElseIf ia_width < 4 * x_ix Then

    ia_width_eff = 3 * x_ix * ia_rails

    ia_res_eff = ix_3x

    ElseIf ia_width < 5 * x_ix Then

    ia_width_eff = 4 * x_ix * ia_rails

    ia_res_eff = ix_4x

    Else

    ia_width_eff = (ia_width * ia_rails) / (5 * x_ix)

    ia_res_eff = ix_5x / ia_width_eff

    End If


Range("E89") = ia_res_eff


ib_width = Range("C90").Value
```

```
ib_rails = Range("D90").Value

    If ib_rails = 0 Then

    ib_res_eff = 1000000

    ElseIf ib_width < 1.5 * x_ix Then

    ib_width_eff = 1 * x_ix * ib_rails

    ib_res_eff = ix_1x

    ElseIf ib_width < 2 * x_ix Then

    ib_width_eff = 1.5 * x_ix * ib_rails

    ib_res_eff = ix_1p5x

    ElseIf ib_width < 2.5 * x_ix Then

    ib_width_eff = 2 * x_ix * ib_rails

    ib_res_eff = ix_2x

    ElseIf ib_width < 3 * x_ix Then

    ib_width_eff = 2.5 * x_ix * ib_rails

    ib_res_eff = ix_2p5x

    ElseIf ib_width < 4 * x_ix Then

    ib_width_eff = 3 * x_ix * ib_rails

    ib_res_eff = ix_3x

    ElseIf ib_width < 5 * x_ix Then

    ib_width_eff = 4 * x_ix * ib_rails

    ib_res_eff = ix_4x

    Else

    ib_width_eff = (ib_width * ib_rails) / (5 * x_ix)

    ib_res_eff = ix_5x / ib_width_eff

    End If
```

```
Range("E90") = ia_res_eff


ma_width = Range("C91").Value

ma_rails = Range("D91").Value

    If ma_rails = 0 Then

    ma_res_eff = 1000000

    ElseIf ma_width < 1.5 * x_mx Then

    ma_width_eff = 1 * x_mx * ma_rails

    ma_res_eff = mx_1x

    ElseIf ma_width < 2 * x_mx Then

    ma_width_eff = 1.5 * x_mx * ma_rails

    ma_res_eff = mx_1p5x

    ElseIf ma_width < 2.5 * x_mx Then

    ma_width_eff = 2 * x_mx * ma_rails

    ma_res_eff = mx_2x

    ElseIf ma_width < 3 * x_mx Then

    ma_width_eff = 2.5 * x_mx * ma_rails

    ma_res_eff = mx_2p5x

    ElseIf ma_width < 4 * x_mx Then

    ma_width_eff = 3 * x_mx * ma_rails

    ma_res_eff = mx_3x

    ElseIf ma_width < 5 * x_mx Then

    ma_width_eff = 4 * x_mx * ma_rails

    ma_res_eff = mx_4x
    Else
    ma_width_eff = (ma_width * ma_rails) / (5 * x_mx)
    ma_res_eff = mx_5x / ma_width_eff
    End If

Range("E91") = ma_res_eff


mb_width = Range("C92").Value
mb_rails = Range("D92").Value
    If mb_rails = 0 Then
    mb_res_eff = 1000000
    ElseIf mb_width < 1.5 * x_mx Then
    mb_width_eff = 1 * x_mx * mb_rails
    mb_res_eff = mx_1x
    ElseIf mb_width < 2 * x_mx Then
    mb_width_eff = 1.5 * x_mx * mb_rails
    mb_res_eff = mx_1p5x
    ElseIf mb_width < 2.5 * x_mx Then
    mb_width_eff = 2 * x_mx * mb_rails
    mb_res_eff = mx_2x
    ElseIf mb_width < 3 * x_mx Then
    mb_width_eff = 2.5 * x_mx * mb_rails
    mb_res_eff = mx_2p5x
    ElseIf mb_width < 4 * x_mx Then
    mb_width_eff = 3 * x_mx * mb_rails
```

```
  mb_res_eff = mx_3x
  ElseIf mb_width < 5 * x_mx Then
  mb_width_eff = 4 * x_mx * mb_rails
  mb_res_eff = mx_4x
  Else
  mb_width_eff = (mb_width * mb_rails) / (5 * x_mx)
  mb_res_eff = mx_5x / mb_width_eff
  End If

Range("E92") = mb_res_eff

hor_res_eff = 1 / ((1 / m2_res_eff) + (1 / m4_res_eff) + (1 / m6_res_eff) + (1 / m8_res_eff) + (1 / b2_res_eff) + (1 / ia_res_eff) + (1 / ma_res_eff))
vert_res_eff = 1 / ((1 / m1_res_eff) + (1 / m3_res_eff) + (1 / m5_res_eff) + (1 / m7_res_eff) + (1 / b1_res_eff) + (1 / ib_res_eff) + (1 / mb_res_eff))
numerator = ((2 * vert_res_eff) + hor_res_eff) * hor_res_eff
denominator = 2 * (vert_res_eff + hor_res_eff)
res_eff = numerator / denominator

Range("F78") = res_eff

End Sub

Sub clampResCalc()
Dim length, width, res_eff, res_eff_1, width_sg_eg As Double
Dim len_eg_1, len_eg_2, area_eg_1, area_eg_2, res_eg_1, res_eg_2, slope_eg As Double
Dim len_sg_1, len_sg_2, len_sg_3, area_sg_1, area_sg_2, area_sg_3, res_sg_1, res_sg_2,
res_sg_3, slope_sg As Double
Dim area As Integer
Dim clamp As String

Range("L42").Select
 Selection.ClearContents
Range("K42").Select
 Selection.ClearContents

Sheets("Placement_Calculator").Select

clamp = Range("H42").Value
length = Range("I42").Value
width = Range("J42").Value
area = length * width
res_sg = 0.25
res_eg = 0.7

Sheets("Technology_Inputs").Select

width_sg_eg = Range("C20").Value

len_sg_1 = Range("C21").Value
```

```
len_sg_2 = Range("D21").Value

len_sg_3 = Range("E21").Value

res_sg_1 = Range("C23").Value

res_sg_2 = Range("D23").Value

res_sg_3 = Range("E23").Value


area_sg_1 = width_sg_eg * len_sg_1

area_sg_2 = width_sg_eg * len_sg_2

area_sg_3 = width_sg_eg * len_sg_3

slope_sg_1 = (res_sg_2 - res_sg_1) / (len_sg_2 - len_sg_1)

slope_sg_2 = (res_sg_3 - res_sg_2) / (len_sg_3 - len_sg_2)


len_eg_1 = Range("F21").Value

len_eg_2 = Range("G21").Value

res_eg_1 = Range("F23").Value

res_eg_2 = Range("G23").Value


area_eg_1 = width_sg_eg * len_eg_1

area_eg_2 = width_sg_eg * len_eg_2

'slope_eg = (res_eg_2 - res_eg_1) / (len_eg_2 - len_eg_1)

slope_eg = (res_eg_2 - res_eg_1) / (area_eg_2 - area_eg_1)


Sheets("Placement_Calculator").Select


If clamp = "SG" And length <= 0.08 And width <> width_sg_eg Then
```

```
    res_eff_1 = slope_sg_1 * (length - len_sg_1) + res_sg_1
    res_eff = (res_eff_1 * width_sg_eg) / width
    Range("K42") = res_eff
    ElseIf clamp = "SG" And length <= 0.08 Then
    res_eff = slope_sg_1 * (length - len_sg_1) + res_sg_1
    Range("K42") = res_eff

    ElseIf clamp = "SG" And length <= 0.12 And width <> width_sg_eg Then
    res_eff_1 = slope_sg_2 * (length - len_sg_2) + res_sg_2
    res_eff = (res_eff_1 * width_sg_eg) / width
    Range("K42") = res_eff
    ElseIf clamp = "SG" And length <= 0.12 Then
    res_eff = slope_sg_2 * (length - len_sg_2) + res_sg_2
    Range("K42") = res_eff
End If

If clamp = "EG" And length <= 0.25 And width <> width_sg_eg Then
    res_eff_1 = slope_eg * (area - area_eg_1) + res_eg_1
    res_eff = (res_eff_1 * width_sg_eg) / width
    Range("K42") = res_eff
    ElseIf clamp = "EG" And length <= 0.25 Then
    res_eff = slope_eg * (area - area_eg_1) + res_eg_1
    Range("K42") = res_eff
End If
```

```
If clamp = "SG" And (length > 0.12 Or length < 0.04) Then
Range("K42") = "Invalid length for SG type"
ElseIf clamp = "EG" And (length > 0.25 Or length < 0.15) Then
Range("K42") = "Invalid length for EG type"
End If

End Sub

Sub hbmUpDiodeResCalc()
Dim length, perimeter, res_eff, ratio, current, current_ref, slope, current_limit, hbm_current,
cdm_current As Double
Dim diode, diode_type, esd_type As String
Dim res_sti_ppnw_0p8, res_sti_ppnw_0p22, res_poly_ppnw_0p14, res_poly_ppnw_0p16,
res_sti_str2_ppnw_0p8, res_sti_str2_ppnw_0p22, res_sti_str3_ppnw_0p8,
res_sti_str3_ppnw_0p22 As Double
Dim res_sti_nppw_0p8, res_sti_nppw_0p22, res_poly_nppw_0p14, res_poly_nppw_0p16 As
Double

Sheets("Placement_Calculator").Select

Range("L18").Select
   Selection.ClearContents
Range("M18").Select
   Selection.ClearContents
Range("N18").Select
```

```
  Selection.ClearContents

current_ref = 13
current_limit = 10

diode = Range("H18").Value
diode_type = Range("I18").Value
length = Range("J18").Value
perimeter = Range("K18").Value

hbm_current = Range("B6").Value
cdm_current = Range("B7").Value
If hbm_current > cdm_current Then
current = hbm_current
Else
current = cdm_current
End If

Sheets("Technology_Inputs").Select

perimeter_sti_ppnw_0p8 = Range("C32").Value
perimeter_sti_ppnw_0p22 = Range("F32").Value
perimeter_sti_nppw_0p8 = Range("D32").Value
perimeter_sti_nppw_0p22 = Range("G32").Value
```

```
res_sti_ppnw_0p8 = Range("C35").Value
res_sti_ppnw_0p22 = Range("F35").Value
res_sti_nppw_0p8 = Range("D35").Value
res_sti_nppw_0p22 = Range("G35").Value

perimeter_poly_ppnw_0p14 = Range("C45").Value
perimeter_poly_ppnw_0p16 = Range("F45").Value
perimeter_poly_nppw_0p14 = Range("D45").Value
perimeter_poly_nppw_0p16 = Range("G45").Value

res_poly_ppnw_0p14 = Range("C48").Value
res_poly_ppnw_0p16 = Range("F48").Value
res_poly_nppw_0p14 = Range("D48").Value
res_poly_nppw_0p16 = Range("G48").Value

perimeter_sti_str2_ppnw_0p8 = Range("C57").Value
perimeter_sti_str2_ppnw_0p22 = Range("E57").Value
perimeter_sti_str3_ppnw_0p8 = Range("D57").Value
perimeter_sti_str3_ppnw_0p22 = Range("F57").Value

res_sti_str2_ppnw_0p8 = Range("C60").Value
res_sti_str2_ppnw_0p22 = Range("E60").Value
res_sti_str3_ppnw_0p8 = Range("D60").Value
res_sti_str3_ppnw_0p22 = Range("F60").Value
```

```
perimeter_0p22_1 = Range("K21").Value
perimeter_0p22_2 = Range("K22").Value
perimeter_0p22_3 = Range("K23").Value
perimeter_0p22_4 = Range("K24").Value
current_ref_ppnw_0p22_1 = Range("L21").Value
current_ref_ppnw_0p22_2 = Range("L22").Value
current_ref_ppnw_0p22_3 = Range("L23").Value
current_ref_ppnw_0p22_4 = Range("L24").Value
current_ref_nppw_0p22_1 = Range("M21").Value
current_ref_nppw_0p22_2 = Range("M22").Value
current_ref_nppw_0p22_3 = Range("M23").Value
current_ref_nppw_0p22_4 = Range("M24").Value
current_ref_ppnw_str_0p22 = Range("O21").Value

perimeter_0p8_1 = Range("K40").Value
perimeter_0p8_2 = Range("K41").Value
current_ref_ppnw_0p8_1 = Range("L40").Value
current_ref_ppnw_0p8_2 = Range("L41").Value
current_ref_nppw_0p8_1 = Range("M40").Value
current_ref_nppw_0p8_2 = Range("M41").Value
current_ref_ppnw_str_0p8 = Range("O40").Value

Sheets("Placement_Calculator").Select

If diode = "STI Diode" And diode_type = "P+/NW" And length = 0.8 Then
```

```
'Range("M19") = perimeter_sti_ppnw_0p8
'Range("M20") = res_sti_ppnw_0p8
'Range("M21") = perimeter
res_eff = perimeter_sti_ppnw_0p8 * res_sti_ppnw_0p8 / perimeter
Range("L18") = res_eff
ElseIf diode = "STI Diode" And diode_type = "P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_ppnw_0p22 * res_sti_ppnw_0p22 / perimeter
Range("L18") = res_eff

ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.14 Then
res_eff = perimeter_poly_ppnw_0p14 * res_poly_ppnw_0p14 / perimeter
Range("L18") = res_eff
ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.16 Then
res_eff = perimeter_poly_ppnw_0p16 * res_poly_ppnw_0p16 / perimeter
Range("L18") = res_eff

ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str2_ppnw_0p8 * res_sti_str2_ppnw_0p8 / perimeter
Range("L18") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str2_ppnw_0p22 * res_sti_str2_ppnw_0p22 / perimeter
Range("L18") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str3_ppnw_0p8 * res_sti_str3_ppnw_0p8 / perimeter
Range("L18") = res_eff
```

```
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str3_ppnw_0p22 * res_sti_str3_ppnw_0p22 / perimeter
Range("L18") = res_eff
Else
Range("L18") = "Invalid combination"
End If
'************determine failure for l=0.22um****************
If diode_type = "P+/NW" And perimeter < perimeter_0p22_2 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_2 - current_ref_ppnw_0p22_1) / (perimeter_0p22_2 - perimeter_0p22_1)
current_limit = slope * (perimeter - perimeter_0p22_1) + current_ref_ppnw_0p22_1
ElseIf diode_type = "P+/NW" And perimeter >= perimeter_0p22_2 And perimeter < perimeter_0p22_3 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_3 - current_ref_ppnw_0p22_2) / (perimeter_0p22_3 - perimeter_0p22_2)
current_limit = slope * (perimeter - perimeter_0p22_2) + current_ref_ppnw_0p22_2
ElseIf diode_type = "P+/NW" And perimeter >= perimeter_0p22_3 And perimeter < perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_4 - current_ref_ppnw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_3) + current_ref_ppnw_0p22_3
ElseIf diode_type = "P+/NW" And perimeter >= perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
```

```
slope = (current_ref_ppnw_0p22_4 - current_ref_ppnw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_4) + current_ref_ppnw_0p22_4
'**************
ElseIf diode_type = "N+/PW" And perimeter < perimeter_0p22_2 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_2 - current_ref_nppw_0p22_1) / (perimeter_0p22_2 - perimeter_0p22_1)
current_limit = slope * (perimeter - perimeter_0p22_1) + current_ref_nppw_0p22_1
ElseIf diode_type = "N+/PW" And perimeter >= perimeter_0p22_2 And perimeter < perimeter_0p22_3 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_3 - current_ref_nppw_0p22_2) / (perimeter_0p22_3 - perimeter_0p22_2)
current_limit = slope * (perimeter - perimeter_0p22_2) + current_ref_nppw_0p22_2
ElseIf diode_type = "N+/PW" And perimeter >= perimeter_0p22_3 And perimeter < perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_4 - current_ref_nppw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_3) + current_ref_nppw_0p22_3
ElseIf diode_type = "N+/PW" And perimeter >= perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_4 - current_ref_nppw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_4) + current_ref_nppw_0p22_4
'***************
```

```
ElseIf perimeter <= perimeter_0p22_4 And length = 0.22 And diode = "STI String Diode" Then
current_lmit = current_ref_ppnw_str_0p22
ElseIf perimeter > perimeter_0p22_4 And length = 0.22 And diode = "STI String Diode" Then
Range("M18") = "Failure criteria is not verified"

'************determine failure for l=0.8um****************
ElseIf diode_type = "P+/NW" And perimeter < perimeter_0p8_2 And length = 0.8 And diode = "STI String Diode" Then
slope = (curret_ref_ppnw_0p8_2 - curret_ref_ppnw_0p8_1) / (perimeter_0p8_2 - perimeter_0p8_1)
current_limit = slope * (perimeter - perimeter_0p8_1) + current_ref_ppnw_0p8_1
ElseIf diode_type = "N+/PW" And length = 0.8 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p8_2 - curret_ref_ppnw_0p8_1) / (perimeter_0p8_2 - perimeter_0p8_1)
current_limit = slope * (perimeter - perimeter_0p8_2) + current_ref_nppw_2
'****************
ElseIf perimeter <= perimeter_0p8_2 And length = 0.8 And diode = "STI String Diode" Then
current_limit = current_ref_ppnw_str_0p8
ElseIf perimeter > perimeter_0p8_2 And length = 0.8 And diode = "STI String Diode" Then
Range("M18") = "Failure criteria is not verified"
End If

If current > current_limit Then
'Range("M19") = current
'Range("M20") = current_limit
```

```
Range("M18") = "Diode does not meet failure criteria"
ElseIf current >= current_ref Then
Range("M18") = "Diode does not meet failure criteria"
End If

If diode = "Poly Diode" Then
Range("M18") = "Not verified for failure criteria"
End If

If diode_type = "N+/PW" And Range("M18").Value <> 0 Then
Range("N18") = "Check is done according to N+/PW(TW) criteria"
End If

End Sub

Sub cdmUpDiodeResCalc()
Dim length, perimeter, res_eff, ratio, current, current_ref, slope, current_limit, hbm_current, cdm_current As Double
Dim diode, diode_type, esd_type As String
Dim res_sti_ppnw_0p8, res_sti_ppnw_0p22, res_poly_ppnw_0p14, res_poly_ppnw_0p16, res_sti_str2_ppnw_0p8, res_sti_str2_ppnw_0p22, res_sti_str3_ppnw_0p8, res_sti_str3_ppnw_0p22 As Double
Dim res_sti_nppw_0p8, res_sti_nppw_0p22, res_poly_nppw_0p14, res_poly_nppw_0p16 As Double
```

```
Sheets("Placement_Calculator").Select

Range("L30").Select
   Selection.ClearContents
Range("M30").Select
   Selection.ClearContents
Range("N30").Select
   Selection.ClearContents

current_ref = 13
current_limit = 10

diode = Range("H30").Value
diode_type = Range("I30").Value
length = Range("J30").Value
perimeter = Range("K30").Value

hbm_current = Range("B6").Value
cdm_current = Range("B7").Value
If hbm_current > cdm_current Then
current = hbm_current
Else
current = cdm_current
End If
```

```
Sheets("Technology_Inputs").Select

perimeter_sti_ppnw_0p8 = Range("C32").Value
perimeter_sti_ppnw_0p22 = Range("F32").Value
perimeter_sti_nppw_0p8 = Range("D32").Value
perimeter_sti_nppw_0p22 = Range("G32").Value

res_sti_ppnw_0p8 = Range("C35").Value
res_sti_ppnw_0p22 = Range("F35").Value
res_sti_nppw_0p8 = Range("D35").Value
res_sti_nppw_0p22 = Range("G35").Value

perimeter_poly_ppnw_0p14 = Range("C45").Value
perimeter_poly_ppnw_0p16 = Range("F45").Value
perimeter_poly_nppw_0p14 = Range("D45").Value
perimeter_poly_nppw_0p16 = Range("G45").Value

res_poly_ppnw_0p14 = Range("C48").Value
res_poly_ppnw_0p16 = Range("F48").Value
res_poly_nppw_0p14 = Range("D48").Value
res_poly_nppw_0p16 = Range("G48").Value

perimeter_sti_str2_ppnw_0p8 = Range("C57").Value
perimeter_sti_str2_ppnw_0p22 = Range("E57").Value
perimeter_sti_str3_ppnw_0p8 = Range("D57").Value
```

```
perimeter_sti_str3_ppnw_0p22 = Range("F57").Value

res_sti_str2_ppnw_0p8 = Range("C60").Value
res_sti_str2_ppnw_0p22 = Range("E60").Value
res_sti_str3_ppnw_0p8 = Range("D60").Value
res_sti_str3_ppnw_0p22 = Range("F60").Value

perimeter_0p22_1 = Range("K21").Value
perimeter_0p22_2 = Range("K22").Value
perimeter_0p22_3 = Range("K23").Value
perimeter_0p22_4 = Range("K24").Value
current_ref_ppnw_0p22_1 = Range("L21").Value
current_ref_ppnw_0p22_2 = Range("L22").Value
current_ref_ppnw_0p22_3 = Range("L23").Value
current_ref_ppnw_0p22_4 = Range("L24").Value
current_ref_nppw_0p22_1 = Range("M21").Value
current_ref_nppw_0p22_2 = Range("M22").Value
current_ref_nppw_0p22_3 = Range("M23").Value
current_ref_nppw_0p22_4 = Range("M24").Value
current_ref_ppnw_str_0p22 = Range("O21").Value

perimeter_0p8_1 = Range("K40").Value
perimeter_0p8_2 = Range("K41").Value
current_ref_ppnw_0p8_1 = Range("L40").Value
current_ref_ppnw_0p8_2 = Range("L41").Value
```

```
current_ref_nppw_0p8_1 = Range("M40").Value
current_ref_nppw_0p8_2 = Range("M41").Value
current_ref_ppnw_str_0p8 = Range("O40").Value

Sheets("Placement_Calculator").Select

If diode = "STI Diode" And diode_type = "P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_ppnw_0p8 * res_sti_ppnw_0p8 / perimeter
Range("L30") = res_eff
ElseIf diode = "STI Diode" And diode_type = "P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_ppnw_0p22 * res_sti_ppnw_0p22 / perimeter
Range("L30") = res_eff

ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.14 Then
res_eff = perimeter_poly_ppnw_0p14 * res_poly_ppnw_0p14 / perimeter
Range("L30") = res_eff
ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.16 Then
res_eff = perimeter_poly_ppnw_0p16 * res_poly_ppnw_0p16 / perimeter
Range("L30") = res_eff

ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str2_ppnw_0p8 * res_sti_str2_ppnw_0p8 / perimeter
Range("L30") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str2_ppnw_0p22 * res_sti_str2_ppnw_0p22 / perimeter
```

```
Range("L30") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str3_ppnw_0p8 * res_sti_str3_ppnw_0p8 / perimeter
Range("L30") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str3_ppnw_0p22 * res_sti_str3_ppnw_0p22 / perimeter
Range("L30") = res_eff
Else
Range("L30") = "Invalid combination"
End If

End Sub

Sub hbmDownDiodeResCalc()
Dim length, perimeter, res_eff, ratio, current, current_ref, slope, current_limit, hbm_current, cdm_current As Double
Dim diode, diode_type, esd_type As String
Dim res_sti_ppnw_0p8, res_sti_ppnw_0p22, res_poly_ppnw_0p14, res_poly_ppnw_0p16, res_sti_str2_ppnw_0p8, res_sti_str2_ppnw_0p22, res_sti_str3_ppnw_0p8, res_sti_str3_ppnw_0p22 As Double
Dim res_sti_nppw_0p8, res_sti_nppw_0p22, res_poly_nppw_0p14, res_poly_nppw_0p16 As Double

Sheets("Placement_Calculator").Select
```

```
Range("L24").Select
   Selection.ClearContents
Range("M24").Select
   Selection.ClearContents
Range("N24").Select
   Selection.ClearContents

current_ref = 13
current_limit = 10

diode = Range("H24").Value
diode_type = Range("I24").Value
length = Range("J24").Value
perimeter = Range("K24").Value

hbm_current = Range("B6").Value
cdm_current = Range("B7").Value
If hbm_current > cdm_current Then
current = hbm_current
Else
current = cdm_current
End If

Sheets("Technology_Inputs").Select
```

```
perimeter_sti_ppnw_0p8 = Range("C32").Value
perimeter_sti_ppnw_0p22 = Range("F32").Value
perimeter_sti_nppw_0p8 = Range("D32").Value
perimeter_sti_nppw_0p22 = Range("G32").Value

res_sti_ppnw_0p8 = Range("C35").Value
res_sti_ppnw_0p22 = Range("F35").Value
res_sti_nppw_0p8 = Range("D35").Value
res_sti_nppw_0p22 = Range("G35").Value

perimeter_poly_ppnw_0p14 = Range("C45").Value
perimeter_poly_ppnw_0p16 = Range("F45").Value
perimeter_poly_nppw_0p14 = Range("D45").Value
perimeter_poly_nppw_0p16 = Range("G45").Value

res_poly_ppnw_0p14 = Range("C48").Value
res_poly_ppnw_0p16 = Range("F48").Value
res_poly_nppw_0p14 = Range("D48").Value
res_poly_nppw_0p16 = Range("G48").Value

perimeter_sti_str2_ppnw_0p8 = Range("C57").Value
perimeter_sti_str2_ppnw_0p22 = Range("E57").Value
perimeter_sti_str3_ppnw_0p8 = Range("D57").Value
perimeter_sti_str3_ppnw_0p22 = Range("F57").Value
```

```
res_sti_str2_ppnw_0p8 = Range("C60").Value
res_sti_str2_ppnw_0p22 = Range("E60").Value
res_sti_str3_ppnw_0p8 = Range("D60").Value
res_sti_str3_ppnw_0p22 = Range("F60").Value

perimeter_0p22_1 = Range("K21").Value
perimeter_0p22_2 = Range("K22").Value
perimeter_0p22_3 = Range("K23").Value
perimeter_0p22_4 = Range("K24").Value
current_ref_ppnw_0p22_1 = Range("L21").Value
current_ref_ppnw_0p22_2 = Range("L22").Value
current_ref_ppnw_0p22_3 = Range("L23").Value
current_ref_ppnw_0p22_4 = Range("L24").Value
current_ref_nppw_0p22_1 = Range("M21").Value
current_ref_nppw_0p22_2 = Range("M22").Value
current_ref_nppw_0p22_3 = Range("M23").Value
current_ref_nppw_0p22_4 = Range("M24").Value
current_ref_ppnw_str_0p22 = Range("O21").Value

perimeter_0p8_1 = Range("K40").Value
perimeter_0p8_2 = Range("K41").Value
current_ref_ppnw_0p8_1 = Range("L40").Value
current_ref_ppnw_0p8_2 = Range("L41").Value
current_ref_nppw_0p8_1 = Range("M40").Value
current_ref_nppw_0p8_2 = Range("M41").Value
```

```
current_ref_ppnw_str_0p8 = Range("O40").Value

Sheets("Placement_Calculator").Select

If diode = "STI Diode" And diode_type = "P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_ppnw_0p8 * res_sti_ppnw_0p8 / perimeter
Range("L24") = res_eff
ElseIf diode = "STI Diode" And diode_type = "P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_ppnw_0p22 * res_sti_ppnw_0p22 / perimeter
Range("L24") = res_eff

ElseIf diode = "STI Diode" And diode_type = "N+/PW" And length = 0.8 Then
res_eff = perimeter_sti_nppw_0p8 * res_sti_nppw_0p8 / perimeter
Range("L24") = res_eff
ElseIf diode = "STI Diode" And diode_type = "N+/PW" And length = 0.22 Then
res_eff = perimeter_sti_nppw_0p22 * res_sti_nppw_0p22 / perimeter
Range("L24") = res_eff

ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.14 Then
res_eff = perimeter_poly_ppnw_0p14 * res_poly_ppnw_0p14 / perimeter
Range("L24") = res_eff
ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.16 Then
res_eff = perimeter_poly_ppnw_0p16 * res_poly_ppnw_0p16 / perimeter
Range("L24") = res_eff
```

```
ElseIf diode = "Poly Diode" And diode_type = "N+/PW" And length = 0.14 Then
res_eff = perimeter_poly_nppw_0p14 * res_poly_nppw_0p14 / perimeter
Range("L24") = res_eff
ElseIf diode = "Poly Diode" And diode_type = "N+/PW" And length = 0.16 Then
res_eff = perimeter_poly_nppw_0p16 * res_poly_nppw_0p16 / perimeter
Range("L24") = res_eff

ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str2_ppnw_0p8 * res_sti_str2_ppnw_0p8 / perimeter
Range("L24") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str2_ppnw_0p22 * res_sti_str2_ppnw_0p22 / perimeter
Range("L24") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str3_ppnw_0p8 * res_sti_str3_ppnw_0p8 / perimeter
Range("L24") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str3_ppnw_0p22 * res_sti_str3_ppnw_0p22 / perimeter
Range("L24") = res_eff
Else
Range("L24") = "Invalid combination"
End If

'************determine failure for l=0.22um****************
```

```
If diode_type = "P+/NW" And perimeter < perimeter_0p22_2 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_2 - current_ref_ppnw_0p22_1) / (perimeter_0p22_2 - perimeter_0p22_1)
current_limit = slope * (perimeter - perimeter_0p22_1) + current_ref_ppnw_0p22_1
ElseIf diode_type = "P+/NW" And perimeter >= perimeter_0p22_2 And perimeter < perimeter_0p22_3 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_3 - current_ref_ppnw_0p22_2) / (perimeter_0p22_3 - perimeter_0p22_2)
current_limit = slope * (perimeter - perimeter_0p22_2) + current_ref_ppnw_0p22_2
ElseIf diode_type = "P+/NW" And perimeter >= perimeter_0p22_3 And perimeter < perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_4 - current_ref_ppnw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_3) + current_ref_ppnw_0p22_3
ElseIf diode_type = "P+/NW" And perimeter >= perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p22_4 - current_ref_ppnw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_4) + current_ref_ppnw_0p22_4
'**************
ElseIf diode_type = "N+/PW" And perimeter < perimeter_0p22_2 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_2 - current_ref_nppw_0p22_1) / (perimeter_0p22_2 - perimeter_0p22_1)
```

```
current_limit = slope * (perimeter - perimeter_0p22_1) + current_ref_nppw_0p22_1
ElseIf diode_type = "N+/PW" And perimeter >= perimeter_0p22_2 And perimeter < perimeter_0p22_3 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_3 - current_ref_nppw_0p22_2) / (perimeter_0p22_3 - perimeter_0p22_2)
current_limit = slope * (perimeter - perimeter_0p22_2) + current_ref_nppw_0p22_2
ElseIf diode_type = "N+/PW" And perimeter >= perimeter_0p22_3 And perimeter < perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_4 - current_ref_nppw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_3) + current_ref_nppw_0p22_3
ElseIf diode_type = "N+/PW" And perimeter >= perimeter_0p22_4 And length = 0.22 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p22_4 - current_ref_nppw_0p22_3) / (perimeter_0p22_4 - perimeter_0p22_3)
current_limit = slope * (perimeter - perimeter_0p22_4) + current_ref_nppw_0p22_4
'***************
Range("N25") = current_limit
'***************
ElseIf perimeter <= perimeter_0p22_4 And length = 0.22 And diode = "STI String Diode" Then
current_lmit = current_ref_ppnw_str_0p22
ElseIf perimeter > perimeter_0p22_4 And length = 0.22 And diode = "STI String Diode" Then
Range("M24") = "Failure criteria is not verified"

'************determine failure for l=0.8um***************
```

```
ElseIf diode_type = "P+/NW" And perimeter < perimeter_0p8_2 And length = 0.8 And diode <> "STI String Diode" Then
slope = (curret_ref_ppnw_0p8_2 - curret_ref_ppnw_0p8_1) / (perimeter_0p8_2 - perimeter_0p8_1)
current_limit = slope * (perimeter - perimeter_0p8_1) + current_ref_ppnw_0p8_1
ElseIf diode_type = "P+/NW" And length = 0.8 And diode <> "STI String Diode" Then
slope = (current_ref_ppnw_0p8_2 - curret_ref_ppnw_0p8_1) / (perimeter_0p8_2 - perimeter_0p8_1)
current_limit = slope * (perimeter - perimeter_0p8_2) + current_ref_nppw_2
'****************
ElseIf diode_type = "N+/PW" And perimeter <= perimeter_0p8_2 And length = 0.8 And diode <> "STI String Diode" Then
slope = (current_ref_nppw_0p8_2 - current_ref_nppw_0p8_1) / (perimeter_0p8_2 - perimeter_0p8_1)
current_limit = slope * (perimeter - perimeter_0p8_1) + current_ref_ppnw_0p8_1
'Range("M27") = slope
'Range("M26") = current_limit
'Range("N19") = perimeter
'Range("N20") = current_ref_nppw_0p8_2
'Range("N21") = current_ref_nppw_0p8_1
'Range("N22") = perimeter_0p8_2
'Range("N23") = perimeter_0p8_1

ElseIf diode_type = "N+/PW" And perimeter > perimeter_0p8_2 And length = 0.8 And diode <> "STI String Diode" Then
```

```
slope = (current_ref_nppw_0p8_2 - curret_ref_nppw_0p8_1) / (perimeter_0p8_2 - perimeter_0p8_1)
current_limit = slope * (perimeter - perimeter_0p8_2) + current_ref_nppw_2
'Range("N24") = "Hello"
'****************
ElseIf perimeter <= perimeter_0p8_2 And length = 0.8 And diode = "STI String Diode" Then
current_limit = current_ref_ppnw_str_0p8
'Range("N25") = "Hello"
ElseIf perimeter > perimeter_0p8_2 And length = 0.8 And diode = "STI String Diode" Then
Range("M24") = "Failure criteria is not verified"
End If

If current > current_limit Then
'Range("M25") = current
'Range("M26") = current_limit
'Range("M27") = slope
Range("M24") = "Diode does not meet failure criteria"
ElseIf current >= current_ref Then
Range("M24") = "Diode does not meet failure criteria"
End If

If diode = "Poly Diode" Then
Range("M24") = "Not verified for failure criteria"
End If
```

```
If diode_type = "N+/PW" And Range("M18").Value <> 0 Then
Range("N24") = "Check is done according to N+/PW(TW) criteria"
End If

End Sub
Sub cdmDownDiodeResCalc()
Dim length, perimeter, res_eff, ratio, current, current_ref, slope, current_limit, hbm_current,
cdm_current As Double
Dim diode, diode_type, esd_type As String
Dim res_sti_ppnw_0p8, res_sti_ppnw_0p22, res_poly_ppnw_0p14, res_poly_ppnw_0p16,
res_sti_str2_ppnw_0p8, res_sti_str2_ppnw_0p22, res_sti_str3_ppnw_0p8,
res_sti_str3_ppnw_0p22 As Double
Dim res_sti_nppw_0p8, res_sti_nppw_0p22, res_poly_nppw_0p14, res_poly_nppw_0p16 As
Double

Sheets("Placement_Calculator").Select

Range("L36").Select
   Selection.ClearContents
Range("M36").Select
   Selection.ClearContents
Range("N36").Select
   Selection.ClearContents

current_ref = 13
```

```
current_limit = 10

diode = Range("H36").Value
diode_type = Range("I36").Value
length = Range("J36").Value
perimeter = Range("K36").Value

hbm_current = Range("B6").Value
cdm_current = Range("B7").Value
If hbm_current > cdm_current Then
current = hbm_current
Else
current = cdm_current
End If

Sheets("Technology_Inputs").Select

perimeter_sti_ppnw_0p8 = Range("C32").Value
perimeter_sti_ppnw_0p22 = Range("F32").Value
perimeter_sti_nppw_0p8 = Range("D32").Value
perimeter_sti_nppw_0p22 = Range("G32").Value

res_sti_ppnw_0p8 = Range("C35").Value
res_sti_ppnw_0p22 = Range("F35").Value
res_sti_nppw_0p8 = Range("D35").Value
```

```
res_sti_nppw_0p22 = Range("G35").Value

perimeter_poly_ppnw_0p14 = Range("C45").Value
perimeter_poly_ppnw_0p16 = Range("F45").Value
perimeter_poly_nppw_0p14 = Range("D45").Value
perimeter_poly_nppw_0p16 = Range("G45").Value

res_poly_ppnw_0p14 = Range("C48").Value
res_poly_ppnw_0p16 = Range("F48").Value
res_poly_nppw_0p14 = Range("D48").Value
res_poly_nppw_0p16 = Range("G48").Value

perimeter_sti_str2_ppnw_0p8 = Range("C57").Value
perimeter_sti_str2_ppnw_0p22 = Range("E57").Value
perimeter_sti_str3_ppnw_0p8 = Range("D57").Value
perimeter_sti_str3_ppnw_0p22 = Range("F57").Value

res_sti_str2_ppnw_0p8 = Range("C60").Value
res_sti_str2_ppnw_0p22 = Range("E60").Value
res_sti_str3_ppnw_0p8 = Range("D60").Value
res_sti_str3_ppnw_0p22 = Range("F60").Value

perimeter_0p22_1 = Range("K21").Value
perimeter_0p22_2 = Range("K22").Value
perimeter_0p22_3 = Range("K23").Value
```

```
perimeter_0p22_4 = Range("K24").Value
current_ref_ppnw_0p22_1 = Range("L21").Value
current_ref_ppnw_0p22_2 = Range("L22").Value
current_ref_ppnw_0p22_3 = Range("L23").Value
current_ref_ppnw_0p22_4 = Range("L24").Value
current_ref_nppw_0p22_1 = Range("M21").Value
current_ref_nppw_0p22_2 = Range("M22").Value
current_ref_nppw_0p22_3 = Range("M23").Value
current_ref_nppw_0p22_4 = Range("M24").Value
current_ref_ppnw_str_0p22 = Range("O21").Value

perimeter_0p8_1 = Range("K40").Value
perimeter_0p8_2 = Range("K41").Value
current_ref_ppnw_0p8_1 = Range("L40").Value
current_ref_ppnw_0p8_2 = Range("L41").Value
current_ref_nppw_0p8_1 = Range("M40").Value
current_ref_nppw_0p8_2 = Range("M41").Value
current_ref_ppnw_str_0p8 = Range("O40").Value

Sheets("Placement_Calculator").Select

If diode = "STI Diode" And diode_type = "P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_ppnw_0p8 * res_sti_ppnw_0p8 / perimeter
Range("L36") = res_eff
ElseIf diode = "STI Diode" And diode_type = "P+/NW" And length = 0.22 Then
```

```
res_eff = perimeter_sti_ppnw_0p22 * res_sti_ppnw_0p22 / perimeter
Range("L36") = res_eff

ElseIf diode = "STI Diode" And diode_type = "N+/PW" And length = 0.8 Then
res_eff = perimeter_sti_nppw_0p8 * res_sti_nppw_0p8 / perimeter
Range("L36") = res_eff
ElseIf diode = "STI Diode" And diode_type = "N+/PW" And length = 0.22 Then
res_eff = perimeter_sti_nppw_0p22 * res_sti_nppw_0p22 / perimeter
Range("L36") = res_eff

ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.14 Then
res_eff = perimeter_poly_ppnw_0p14 * res_poly_ppnw_0p14 / perimeter
Range("L36") = res_eff
ElseIf diode = "Poly Diode" And diode_type = "P+/NW" And length = 0.16 Then
res_eff = perimeter_poly_ppnw_0p16 * res_poly_ppnw_0p16 / perimeter
Range("L36") = res_eff

ElseIf diode = "Poly Diode" And diode_type = "N+/PW" And length = 0.14 Then
res_eff = perimeter_poly_nppw_0p14 * res_poly_nppw_0p14 / perimeter
Range("L36") = res_eff
ElseIf diode = "Poly Diode" And diode_type = "N+/PW" And length = 0.16 Then
res_eff = perimeter_poly_nppw_0p16 * res_poly_nppw_0p16 / perimeter
Range("L36") = res_eff

ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.8 Then
```

```
res_eff = perimeter_sti_str2_ppnw_0p8 * res_sti_str2_ppnw_0p8 / perimeter
Range("L36") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "2STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str2_ppnw_0p22 * res_sti_str2_ppnw_0p22 / perimeter
Range("L36") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.8 Then
res_eff = perimeter_sti_str3_ppnw_0p8 * res_sti_str3_ppnw_0p8 / perimeter
Range("L36") = res_eff
ElseIf diode = "STI String Diode" And diode_type = "3STR P+/NW" And length = 0.22 Then
res_eff = perimeter_sti_str3_ppnw_0p22 * res_sti_str3_ppnw_0p22 / perimeter
Range("L36") = res_eff
Else
Range("L36") = "Invalid combination"
End If

End Sub
```

References

[1] Pinkesh Shah, Prasan Shanbhag, Rahul Shelke, Gargee Das, “A Staggered CUP IO methodology using Encounter platform,” Cadence Design Systems, San Jose, CA, Oct 11, 2007.

[2] Stephen M. Nolan and Jose M. Soltero, “Understanding and Interpreting Standard-Logic Data Sheets,” Texas Instruments, Dallas, TX, SZZA036B, May, 2003

[3] Dr. Yong-Bin Kim (2011). Layout Design Examples [Online]. Available: http://www.ece.neu.edu/courses/eece4525/2011fa/Lab3/Layout_Examples.pdf (URL)

[4] Karan Bhatia and Elyse Rosenbaum, “Layout Guidelines for Optimized ESD Protection Diodes,“ EOS/ESD symposium 07-19.

[5] Oleg Semenov, Hossein Sarbishaei, Manoj Sachdev, “ESD devices for input/output protection” in

ESD Protection Devices and Circuit Design for Advanced CMOS Technologies, Canada, Springer, 2008, ch. 2, pp 45-71.

[6] J. P. Uyemura, “The CMOS inverter: analysis and design” in CMOS Logic Circuit Design. Boston, Kluwer Academic Publishers, 2002, ch. 3, pp 103-139.

[7] H. G. Feng, K. Gong, R. Y. Zhan and Albert Z. Wang, “On-Chip ESD Protection Design for ICs”, Integrated Electronics Laboratory, Dept. of Electrical and Computer Engineering, Illinois Institute of Technology.

[8] K. Banerjee, A. Amerasekera, G. Dixit, N. Cheung, and C. Hu, “Characterization of Contact and Via Failure under Short Duration High Pulsed Current Stress,” in Proceedings of International Reliability Physics Symposium, pp. 216–220, 1996.

[9] M. D. Chaine, C. T. Liong, and H. F. San, “A Correlation Study Between Different Types of CDM Testers and Real Manufacturing In-Line Leakage Failures,” IEEE Trans. Components,

Packaging, and Manufacturing Tech. Part A. Vol. 18, No. 2, pp. 295–302, June, 1995.

[10] ESD Association WG 5.1, “ESD STM5.1-2001: Standard Test Method for Electrostatic Discharge Sensitivity Testing- Human Body Model (HBM) Component Level,” 2001.

[11] S. Voldman, ESD Physics and Devices: John Wiley & Sons, Ltd., 2004.

[12] Jung-Hoon Chun, ”ESD protection circuits for advanced CMOS technologies,” Ph.D. dissertation, Dept. of Electrical Engineering, Stanford University, Stanford, CA, 2006.

[13] M. Kelly, T. Diep, S. Twerefour, G. Servais, D. Lin, G. Shah, “A Comparison of Electrostatic Discharge Models and Failure Signatures for CMOS Integrated Circuit Devices”, EOS/ESD symposium 95-185.

[14] STEPHEN G. BEEBE , “Characterization, Modeling, and Design of ESD Protection Circuits,” Ph.D. dissertation, Dept. of Electrical Engineering, Stanford University, Stanford, CA, 1998.

[15] IBIS Cookbook

Biographical Information

Shiju Abraham is a candidate for the Master of Science degree at The University of Texas at Arlington. He received his Bachelor's degree in Electronics and Communication from Cochin University of Science and Technology in 2007. He worked with Wipro Technologies (Semiconductor and Systems department) for about 4.5 years. He was responsible for STD/IO cell library development, IBIS modeling and top-level place and route of ICs while working with Wipro Technologies. He joined UT Arlington in August 2012 and focused on VLSI/IC design courses during his study. He did a Co-op with GLOBALFOUNDRIES where he worked with the IO/ESD team of the company, on GPIO design. His interests are timing, power analysis on cell level as well as full chip level, interface circuit design, ESD related activities and automatic place and route. He will join Freescale Semiconductor as IP Design Engineer by July 2014. Please feel free to connect with me using shiju.abraham@mavs.uta.edu or http://www.linkedin.com/in/shijuabraham13 .