Output Power Maximization in Energy Harvesting Applications

Dissertation

Submitted in partial fulfillment of the requirements for

the degree of

Bachelor of Technology

by

Nitin Kamra

Entry No: 2010EE10465

under the guidance of

Prof. Shouribrata Chatterjee

Department of Electrical Engineering

Indian Institute of Technology, Delhi

2014

Dissertation Approval Certificate

Department of Electrical Engineering
Indian Institute of Technology, Delhi

The dissertation entitled "**Output Power Maximization in Energy Harvesting Applications**", submitted by **Nitin Kamra** (Entry No: 2010EE10465) is approved for the award of **B. Tech in Electrical Engineering** from **Indian Institute of Technology, Delhi.**

Declaration

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature: ________________________________

Name: **Nitin Kamra**

Entry No: **2010EE10465**

Date: ________________________________

Abstract

The project focuses on increasing the efficiency of an existing Energy Harvesting Integrated Circuit architecture developed by Prof. Shouri Chatterjee's team in the ICE group at IIT Delhi. The project consists of two parts, each of which was undertaken in a separate semester long duration.

The first part was an exploratory step taken in order to explore the possibility of using a Discrete Time Parametric Amplifier (DTPA) for boosting the charging speed of a DC-DC converter, in the Energy Harvesting IC. The DC-DC converter was explored & the functioning of the DTPA was thoroughly analyzed along with various simulations. Then an effort was made to integrate the DTPA with the existing DC-DC converter circuit in order to gain a faster clocking rate & hence increase the charging speed of the output capacitor. In the end though, the idea didn't work out & it was mathematically proved that the DTPA can't give any appreciable gains in the performance of the Harvesting circuit, rather it would only lead to a further deterioration.

The second part focuses on maximum power point tracking for the Energy Harvesting IC. It involved developing an algorithm to track the frequency of charging cycles which would lead us to harness the maximum output power from the Energy source. This involved the development of a digital controller to implement the algorithm on-chip. Further, since the algorithm had lots of idle time in between various charge cycles, the algorithm was improved to incorporate charging from three different sources in parallel while sharing the maximum possible infrastructure & carefully arbitrating between the charging cycles of the three independent sources. The algorithm was successfully demonstrated with simulations in Virtuoso Analog Design Environment.

Acknowledgements

I would like to take this opportunity to express my gratitude to my B.Tech project supervisor Prof. Shouribrata Chatterjee, who originally suggested me this topic for my Major Project and guided me all throughout the course of the project. I am sincerely grateful to him for his excellent guidance towards this cause, for he always posed leading questions to me and helped me build a thorough understanding of the topic. Secondly, I want to thank Prof. G.S. Visweswaran for his invaluable suggestions which helped me a lot, whenever I got stuck anywhere during the project.

I would also like to thank Dr. Gajendranath Chowdary (PhD student) for all his help without which finishing the project would not have been possible on time. Lastly, I thank my family for their constant encouragement & support which inspired me towards the completion of this project.

Nitin Kamra

Table of Contents

1. Introduction

This chapter starts by first introducing the readers to the idea behind energy harvesting and how it can be useful in the current world of electronics. It further goes down into a brief description of the basic DC-DC converter which is at the core of the Energy Harvesting IC and around whose performance the whole Harvesting process revolves. An intuitive explanation of the working has been provided for the reader to understand how the core of the device works. Later several simulation results have been shown which demonstrate the explained theory.

1.1. Energy Harvesting

Energy Harvesting is the process by which energy is derived from external sources (solar, thermal, wind, RF etc.), captured & stored for small, wireless & autonomous devices (1). Energy Harvesting Systems capture energy from ambient sources & transform it into a usable form for electronic gadgets. In general, they provide a very small amount of power which is only feasible for use in low-energy electronics such as wearable electronics & wireless sensor networks (1).

The real idea behind using the technology is that there is lots of ambient energy in the background, which can be exploited to power-up small systems & make them self-sufficient in terms of power. This makes them highly useful to install at places where providing a power supply is difficult or changing their batteries periodically is not feasible. The advantage offered by harvesting is that all this ambient energy is for free & would have anyways been wasted, if not harnessed.

There have been many attempts at harnessing energy from many kinds of sources. The current architecture focuses on harvesting energy from the RF signals that are present in our ambient environment.

1.2. How to exploit this energy?

Having said that we can exploit this ambient energy around us, the question is how this can be accomplished. This is typically done by first using an appropriate energy transducer to convert the energy from its ambient form to an electrical form say, voltage or current. The transducer to be used can differ according to the source from which the energy is being harnessed. For instance, photovoltaic cells can be used to convert solar energy into electrical current. Similarly, piezoelectric substances can sense pressure variations caused by mechanical vibrations and convert them to electrical voltage.

Once the energy has been transformed to the electrical domain, the obtained signal in general needs processing. In most cases, the obtained electrical signal would be only a few tens or hundreds of millivolts, which is not capable of driving any electronic gadgets. This means that the signal must be amplified before it can be put to any good use. For this purpose, we cannot use conventional amplifiers since they need a lot of external power to amplify signals, the luxury of which we do not have for energy harvesting applications. Hence we use a DC-DC boost converter which amplifies the signal using mostly passive components in general. The small amount of energy required for switching purposes & the amount wasted in losses can be obtained from the energy produced from the processed signal. Hence, the harvesting process can be sustained. Next I shall describe the structure & working of the basic DC-DC boost converter since it is central to the IC that I have been working on.

1.3. Structure & Working of the basic DC-DC converter

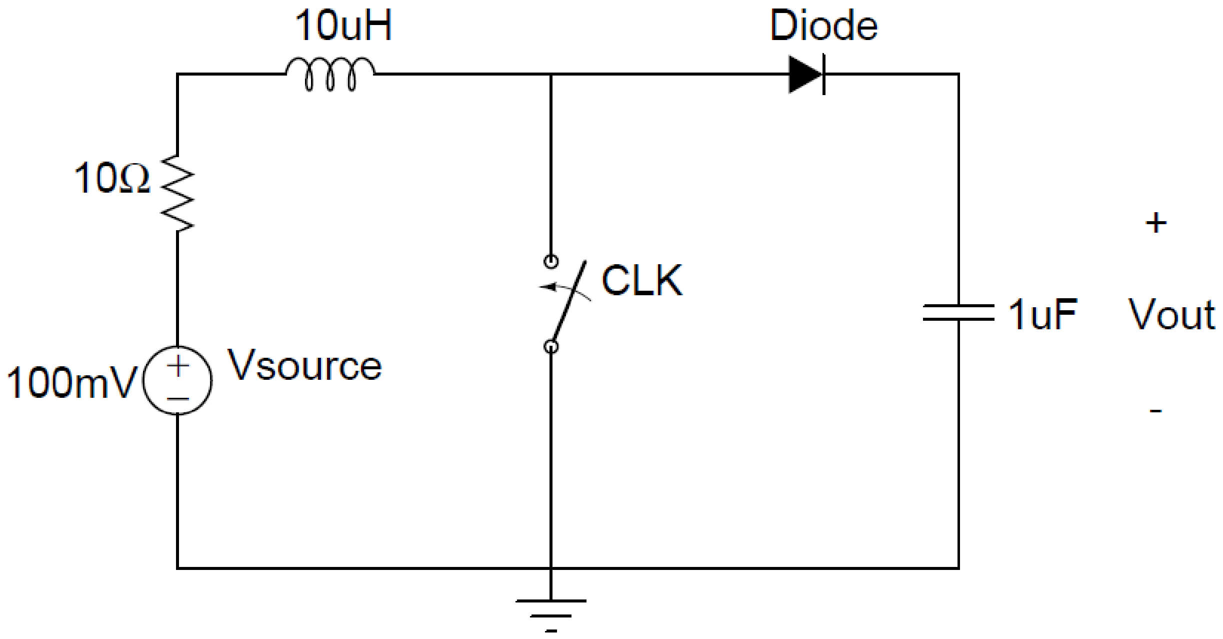

Figure 1: DC-DC converter basic circuit

Figure 1 above shows the circuit diagram of a basic DC-DC converter. On the left, we have a DC energy source which can typically give out some voltage b/w 20mV to 400mV, depending upon the availability of ambient energy in its surrounding environment. The 10Ω resistor models the internal impedance of this energy source. The source is further connected to an inductor which can be grounded at the other end with a switch, controlled with a clock signal. If left ungrounded, the inductor connects to a capacitor through a diode. The voltage stored on the capacitor can be taken as the output voltage & be used to power-up devices.

The circuit works as follows:

- When the CLK signal is high, the switch is closed & the inductor builds up current.
- When the CLK signal goes low, the current in the inductor discharges into the capacitor, thereby increasing the charge on it.
- The diode (assumed ideal for now), cuts off the current flowing into the capacitor, if it tends to reverse its direction & hence prevents any back-flow of charge from the capacitor.
- The inductor is needed to push current into the capacitor, since otherwise after the capacitor voltage increases beyond the terminal voltage of the battery, the diode will cease to conduct & the capacitor won't charge any further.

1.4. Simulations on the basic DC-DC converter

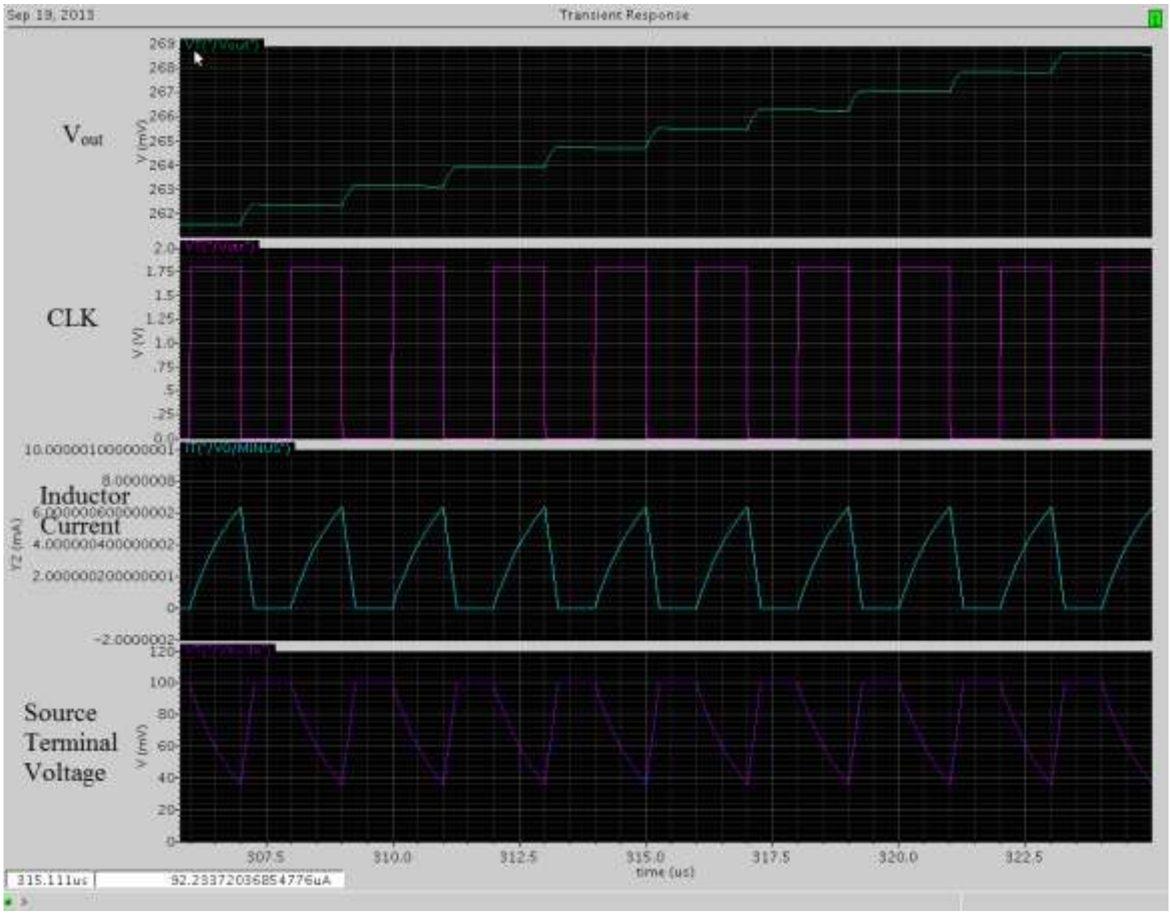

Figure 2: Basic DC-DC simulation results

Many simulations were performed on the basic DC-DC converter circuit to understand its functioning properly. Figure 2 below shows the results of a simulation, which verifies the above explained working.

On every positive clock level, the inductor current increases & consequently, the battery terminal voltage drops. When the clock signal goes to zero, the inductor discharges its current through the capacitor, until the diode cuts off. This leads to a step increase in the charge on the capacitor.

1.5. Adding an input shunt capacitor to the basic DC-DC circuit

It can be seen in the above graphs, that the voltage at the battery terminal is highly fluctuating. But, it is known that maximum power can be derived from a battery, if its terminal voltage is half of its internal EMF. So, next a shunt capacitor was placed across the battery to stabilize the terminal voltage of the battery. This capacitor would hold charge & not let the voltage drop immediately. A 10kΩ resistor was also placed at the output terminal in order to model the effects of loading the DC-DC circuit.

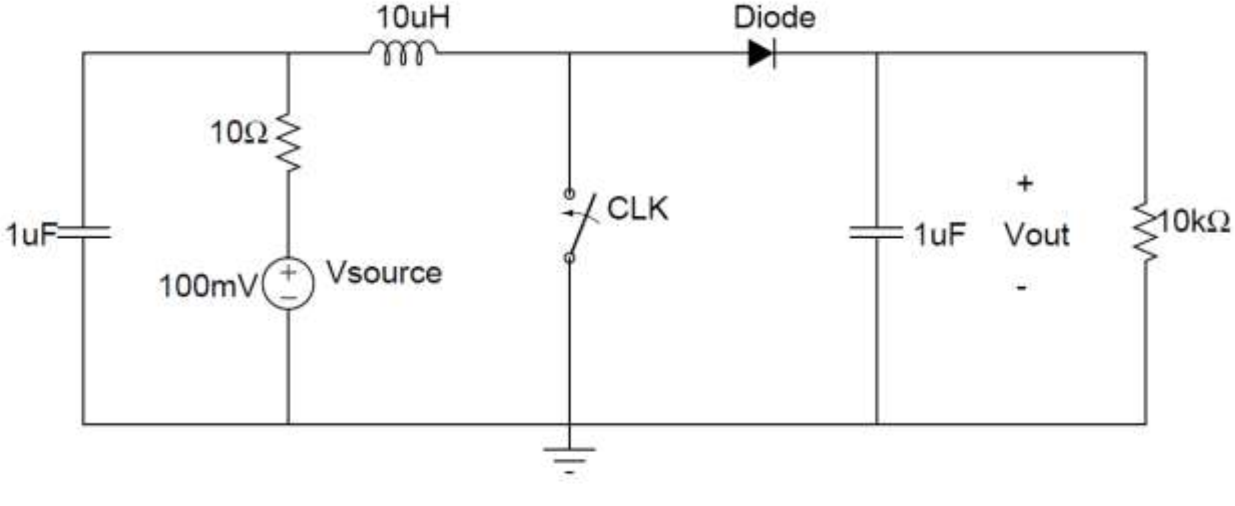

Figure 3: Improved DC-DC converter

The circuit works on the same principle, but this time the shunt capacitor prevents rapid fluctuations of the battery terminal voltage & allows us to extract more power out of the circuit.

1.6. Simulations on the Improved DC-DC converter circuit

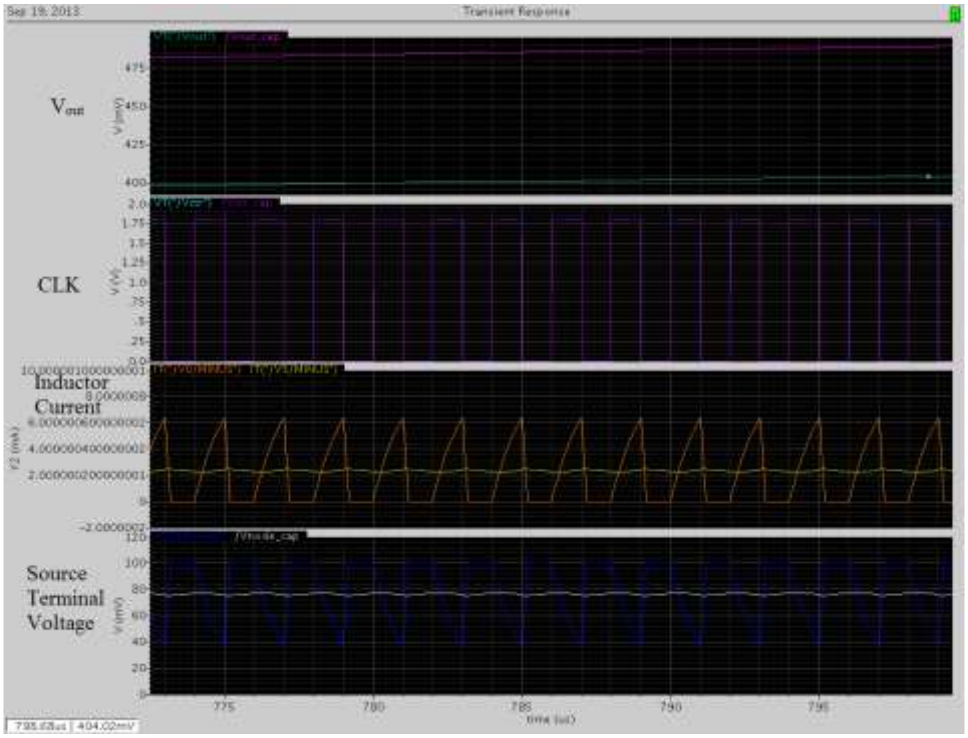

Figure 4: Improved DC-DC Simulation results

The graph in figure 4 above compares the waveforms obtained for the basic DC-DC circuit to the improved one, both with the same load of 10kΩ. It can be seen that the heavy fluctuations of the battery terminal voltage have been effectively suppressed & the voltage waveforms for both the inductor current & the battery terminal voltage are consequently a lot smoother. Though, this doesn't make the battery terminal voltage equal to half the internal EMF, but it still stabilizes the voltage to some point nearby & hence, improves the power transferred to the load.

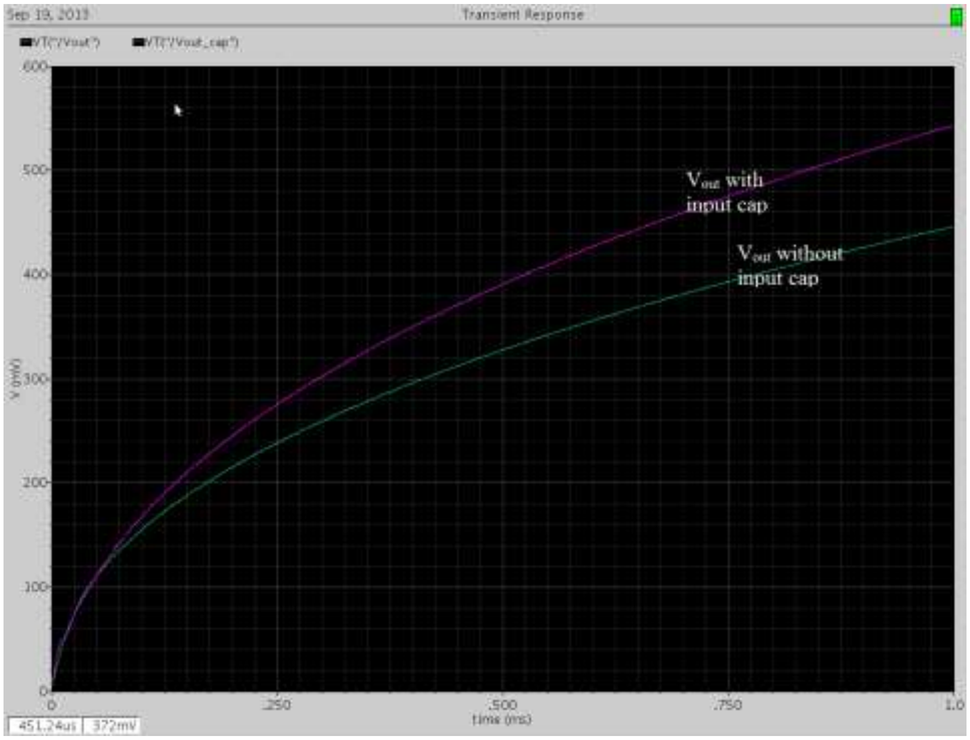

Figure 5: Comparison of Output voltage

Figure 5 shows only the output voltage waveforms for the 2 cases & it can be inferred that the output voltage is always higher for the Improved DC-DC circuit.

Figure 6 below compares the waveforms for output current into the load for the basic DC-DC circuit & the improved DC-DC circuit. Similar observations as for voltage can be made regarding the current too. The improved DC-DC circuit always sends more current to the load resistance. As a consequence of both voltage & current being higher, the power supplied to the load is also higher in case of the Improved DC-DC circuit. This has been shown in Figure 7.

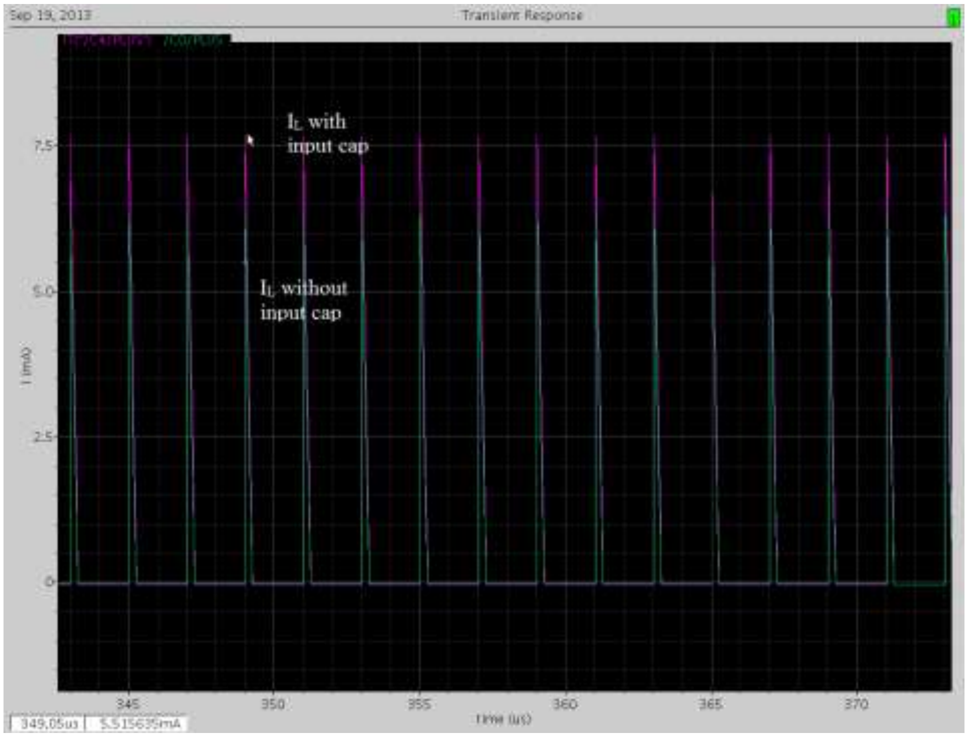

Figure 6: Comparison of Output current

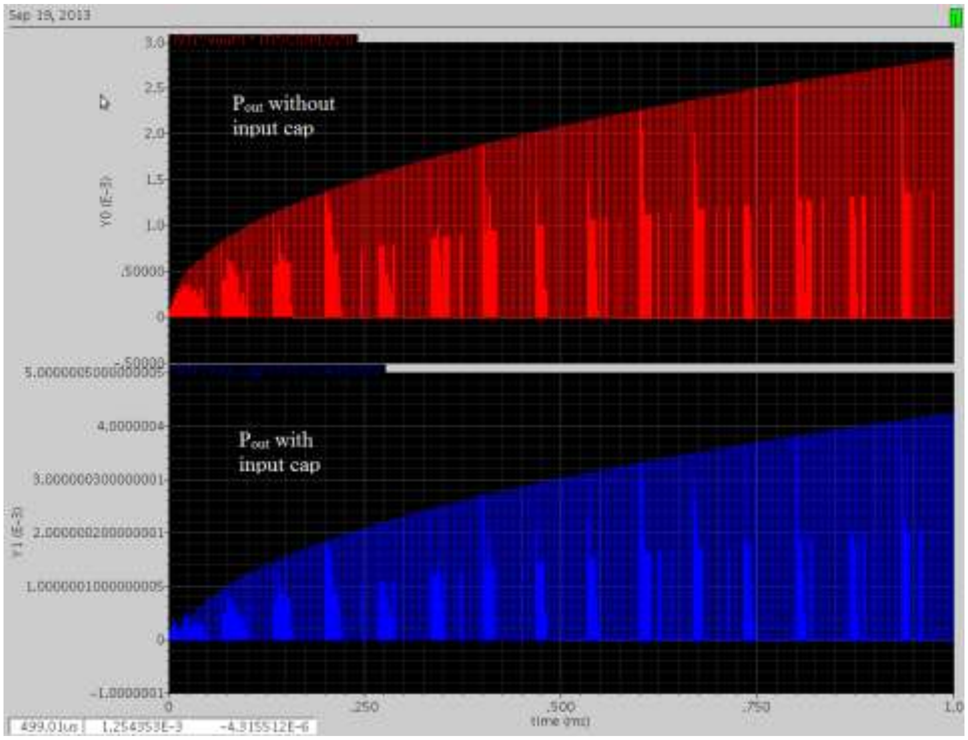

Figure 7: Comparison of Output power

2. The Discrete-Time Parametric Amplifier (DTPA)

2.1. Principle behind the DTPA

The discrete time parametric amplifier is, as the name suggests, an amplifier based on the principle of a variable capacitance, i.e. it exploits the equation: $Q = C * V$ in order to boost voltages. It involves the use of a three-terminal MOS varactor & produces amplification as follows:

- **Track Phase**: The DTPA first tracks a voltage on the capacitor & stores it (2).
- **Hold Phase**: Next, it cuts off the varactor from the supply in order to hold the charge isolated from the supplying source (2).
- **Boost Phase**: Now, the capacitance of the varactor is decreased while holding the charge on it constant. This leads to a boost in the capacitor voltage with the output voltage being V_O (2).

$$V_O = \left(\frac{C_I}{C_O}\right) * V_I \qquad [1]$$

The operating principle has been illustrated in figure 8 below:

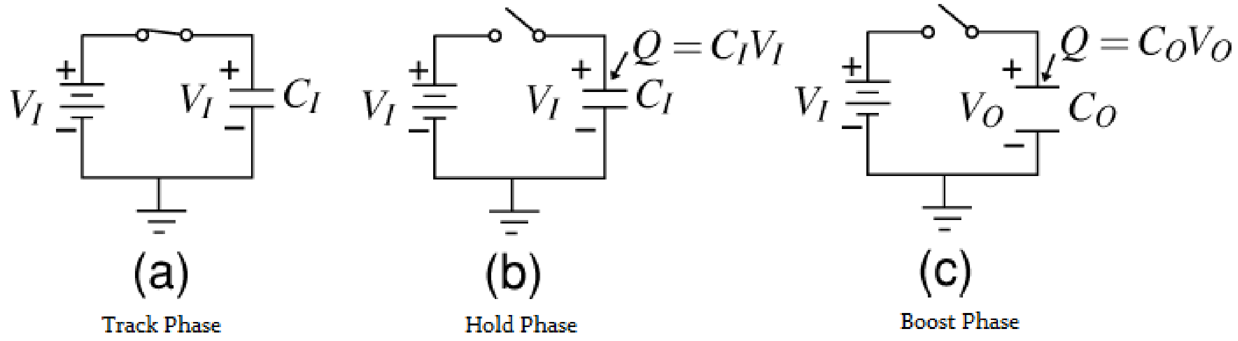

Figure 8: Three phases of a DTPA (2)

2.2. Structure & Working of the DTPA

The implementation of the three-terminal varactor of the DTPA is achieved with a four-terminal MOS transistor (2). Using an NMOS, we short the drain & the source together and ground the body of the NMOS. The D-S terminals can be connected to either ground or V_{DD} through a controlled switch. The gate of the NMOS is connected to the source through another switch which is controlled through a CLK waveform.

2.2.1. Track Phase

Figure 9 shows the cross-section of the DTPA during its track phase. The switch connected to the Gate Terminal is kept closed, thereby shorting it to the voltage source & the D-S terminals are grounded (2). Hence, during this phase the NMOS acts as a capacitor & tracks the input voltage on its Gate Terminal.

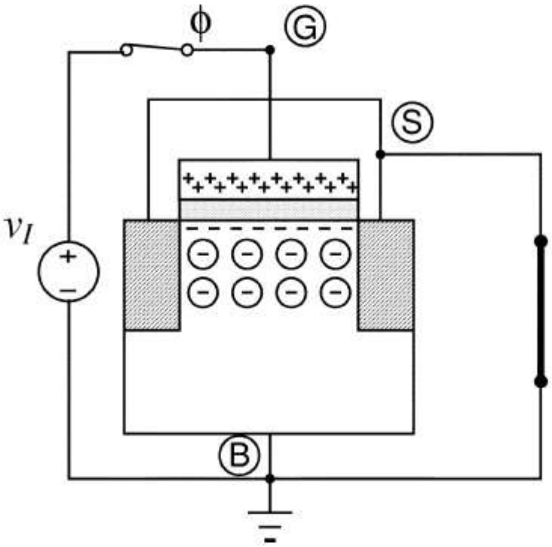

Figure 9: Track Phase of the DTPA (2)

2.2.2. Hold Phase

Now, the switch at the Gate Terminal is opened (2). The Gate Terminal is now isolated & the input charge sampled on it will remain conserved (2). The input voltage is assumed sufficient to be able to create a strong inversion layer. Figure 10 shows the hold phase of the DTPA.

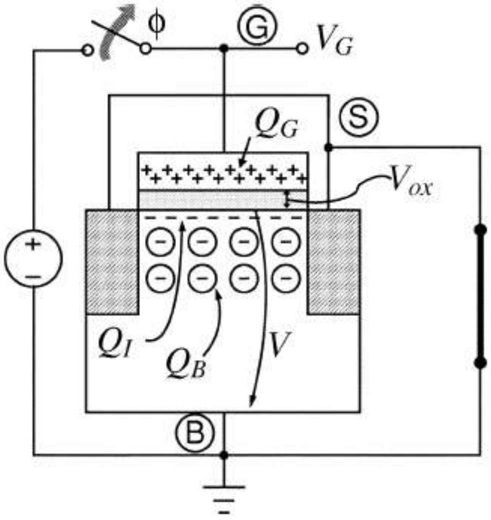

Figure 10: Hold Phase of the DTPA (2)

Let the isolated gate charge be Q_G, the inversion layer negative charge be Q_I & the negative charge exposed in the body be Q_B. Then from charge neutrality, we know that:

$$Q_B = |Q_I| + |Q_B| \qquad (2)[2]$$

Then the gate voltage is given by:

$$V_G = V_{OX} + V \qquad (2)[3]$$

where V_{OX} is the potential drop across the oxide & V is the surface potential drop i.e. the potential drop from the oxide-semiconductor interface to the bulk.

The potential drop across the oxide is given by:

$$V_{OX} = \frac{Q_G}{C_{OX}} \qquad (2)[4]$$

where C_{OX} is the oxide capacitance. Thus, it can be seen that V_{OX} is only a function of the Gate charge. The potential drop across the bulk (V) is a monotonically increasing function of $|Q_B|$. It turns out that:

$$V \propto |Q_B|^2 \qquad (3)[5]$$

2.2.3. Boost Phase

During boosting operation, the D-S terminal is connected to a large positive voltage V_{PULL}, which drains out all the inversion charge out of the NMOS channel. Since, the gate is floating, its charge remains the same & hence must now be balanced by the charge exposed in the body of the NMOS transistor i.e. $Q_I = 0$ & $|Q_B| = Q_G$ (2). Hence, V_{OX} still remains the same, but V being proportional to $|Q_B|^2$ increases tremendously, thus the gate voltage V_G also gets boosted (2).

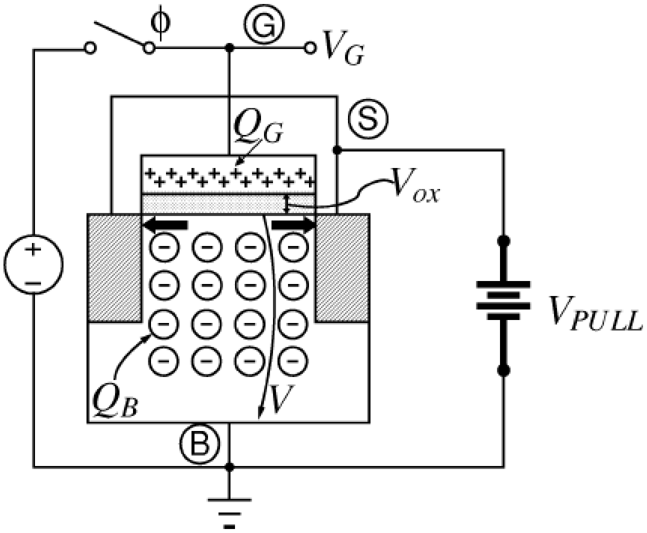

Figure 11: Boost Phase of DTPA [2]

2.3. Differential Operation of DTPA

Since in the above operation of the DTPA, there is no way to distinguish between the bias voltage & the signal voltage, we can also use the DTPA in a differential topology in order to observe the small signal gain of the amplifier.

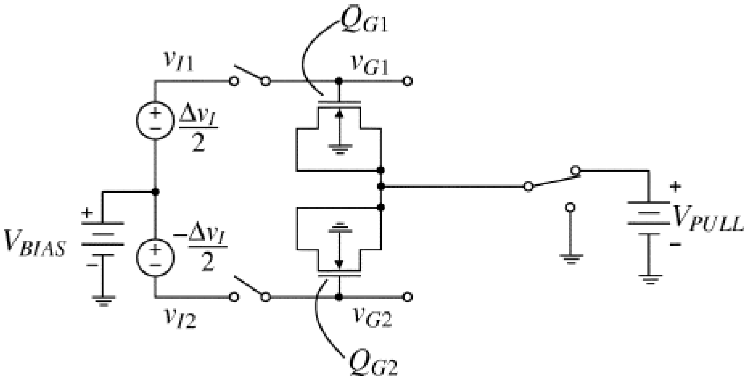

Figure 12: Differential Operation of DTPA (2)

The AC gain can be obtained as:

$$\Delta V_O \ = \left(\frac{C_{OX}}{C_{GB}}\right) * \Delta V_I \qquad (2)\ [6]$$

2.4. Simulation results for Differential Operation of DTPA

The simulation on DTPA was done on both the single-ended & differential topology. For the sake of brevity, only the results from the differential operation are being presented here. The AC gain value found was ≈ 6. The gain waveform is shown below in Figure 13.

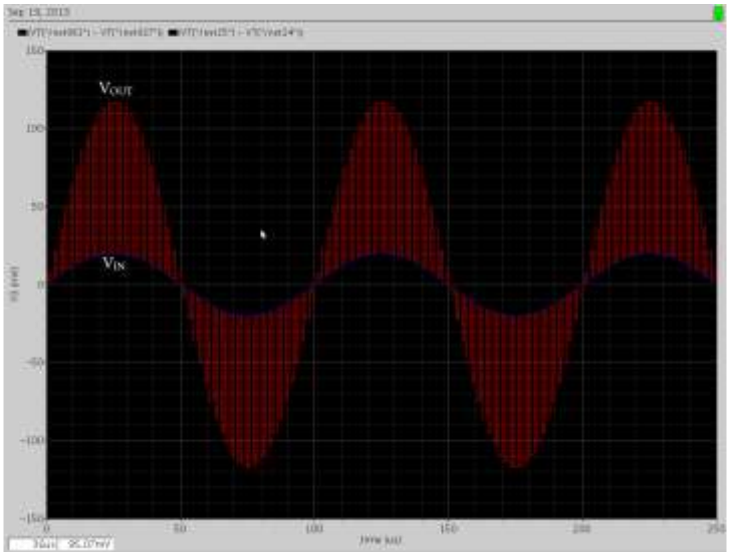

Figure 13: Differential DTPA simulation result

There are some points to be cautious about, though:

- The DTPA needs a long channel length to work. I used 4.2um length & width for the simulation. With 180nm channel length, only unity gain is observed, since there is no AC boosting but only a DC level shift in the input signal.
- AC gain is highly sensitive to DC level of the output signal. So, keep $V_{BIAS} \in$ [300mV, 700mV] to obtain an acceptable AC gain.
- More parameters for the simulation are as follows: CLK frequency = 500 KHz, Input signal amplitude = 20mV peak-peak, Input signal frequency = 10 KHz.
- Table 1 below lists the AC gain obtained as the bias voltage was varied:

DC bias (in mV)	200	300	400	500	600	700	800
AC gain (approx.)	1.08	1.8	4.2	5	5.5	6	1.08

Table 1: AC gain variation with DC bias

3. Integrating the DTPA with DC-DC converter

3.1. Using a DTPA to increase charging speed

In this part, I have tried to explore the possibility if integrating a Discrete Time parametric amplifier with the DC-DC converter to boost the overall charging speed of the circuit. The DTPA offers certain advantages over other kinds of amplifiers such as no resistive losses, very high bandwidth & it is theoretically 100% efficient in terms of energy transfer. Such features allow it to boost the charging voltage with almost negligible losses & hence, if this feature can be exploited for charging the battery, the clocking period of the circuit will go down, thereby leading to a faster charging speed.

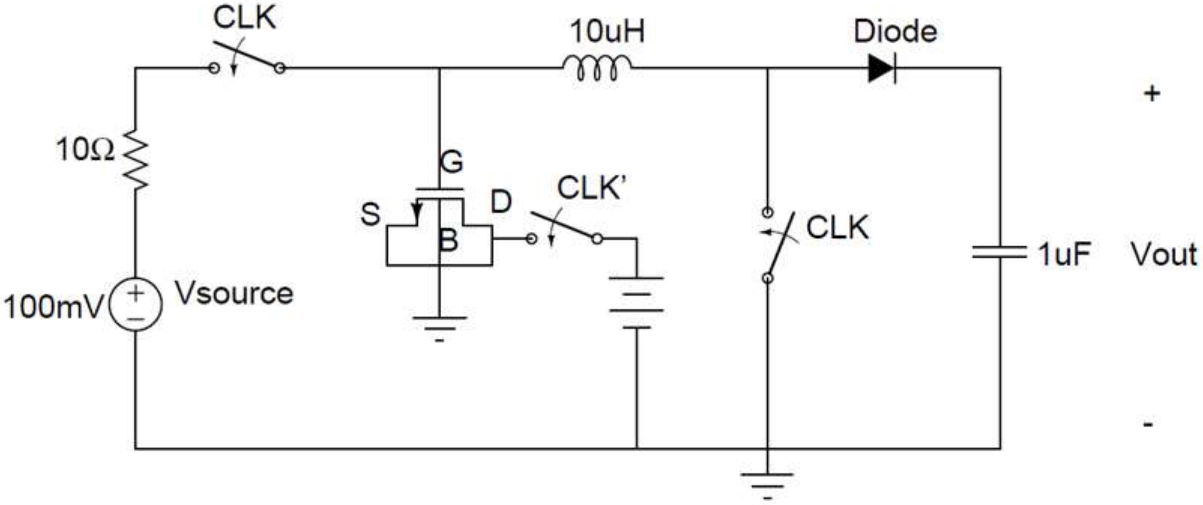

Figure 14: DTPA integrated with DC-DC converter

The idea while integrating the DTPA with the DC-DC converter is to replace the capacitor at the battery terminals with the DTPA-varactor. When the clock signal is high, the inductor current is being built & the DTPA is simultaneously tracking the input voltage. When the clock goes low, both the input switch & the switch shorting the inductor to ground, are opened simultaneously, while a complementary clock signal at the D-S Terminal of the DTPA causes the Gate voltage to get boosted (while maintaining the charge at the Gate Terminal nearly constant). After this the DTPA pushes current into the output capacitor.

Note that even if we had used a passive capacitor in place of the DTPA, the amount of charge which can be pushed onto the output capacitor, after cutting-off from the input source, is same as before. The benefit DTPA is expected to provide is that, after boosting, it has a higher voltage & a lower capacitance. This means it can push the same charge onto the output capacitor much faster than an ordinary passive capacitor without boosting, would have pushed. So, the charging cycle of the output cap is faster. Consequently, we can either change the duty cycle of our controlling clocks (while keeping the same clock frequency) & give more time for the charging of the inductor, which would lead to more current being developed in the inductor OR we can increase the clock frequency & hence, the charge-discharge cycles would be faster than they previously were. Either of these would lead to a faster charging of the output capacitor.

3.2. Simulation & Testing

The above schematic was simulated & some problems were identified. Many of them were corrected, but in the end one of the problems led me to realizing that this operation cannot work. First, I'll present some problems which were sorted easily:

- This time when the inductor builds up current, we need a higher input voltage to account for the bias voltage needed in order to create a strong inversion in the DTPA varactor. For the purpose of simulations a DC-bias source was added in series with the actual source to get a net input voltage of about 600mV, though this must be sorted out, since such a source is not available for the actual circuit of the RF energy harvester. This can be sorted out by using a PMOS based DTPA instead of the one, which uses an NMOS.
- When the clock goes low, & boosting occurs at the DTPA Gate Terminal, it should be noted that this time, the Gate Terminal is not really isolated during boosting, so the charge is not conserved. Some charge from the Gate Terminal escapes to the output capacitor during the process of boosting, but it is acceptable, since we anyways need to push the charge on the DTPA into the output cap.
- Moreover, the clocks need to be timed precisely in the form of break-before-make circuit i.e. non-overlapping clocks in order to get the boosted gain, else the charge on the Gate of the DTPA might flow into ground.
- Further, the circuit was carefully designed so that the inductor always has a path for its current to flow. Since, if at any point of time, there is no path for the inductor current to flow, it will undergo a heavy discharge & lose all its stored energy.
- A single DTPA has very less C_{GB} (Gate-to-body capacitance) & hence can store very less charge alone. This means that very small amount of charge would be available to be pushed into the output cap after boosting. This can be rectified, by putting many DTPAs in parallel in order to add the capacitances.
- The diode can be implemented using a MOS switch controlled by a Comparator, which is comparing the Drain & Source terminal voltages of the MOS switch to prevent backflow of current.
- Lastly, till now we have implemented all the clock signals as external batteries which are not actually present in the DC-DC converter. In reality, these will be

implemented with oscillators drawing power from the output capacitor itself in order to implement the switching in the whole circuit.

3.3. The biggest Problem

The last problem in the above list turned out to be the biggest problem & eventually led to the proof that the circuit can't work. If it is only the switches & the comparator to be powered, then the design can be optimized to take care of these needs. But, if the charge needed to switch the boosting clock on/off also needs to come from the capacitor, then it can be shown that we'll always be drawing charge from the output cap each cycle on a whole, instead of charging it.

3.4. Mathematical Analysis of the problem

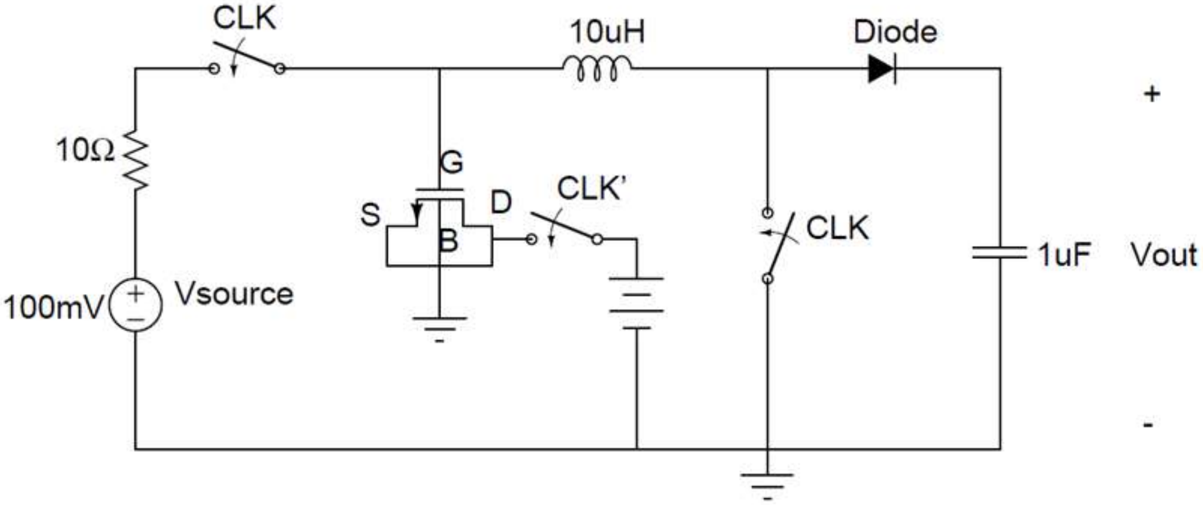

Figure 15: DTPA with DC-DC

Let the input source voltage be denote by V_S. Then the charge pushed onto the DTPA during track phase is given by: V_S*C_{GS}. As only a fraction of this charge gets pushed onto the output capacitor when the clock goes low, let the charge pushed on the output cap per clock cycle be:

$$Q_{GAIN} = \mu * V_S * C_{GS}; \ \mu \epsilon [0, 1] \quad [7]$$

Though μ∈[0, 1] its typical values actually range from [0.4, 0.7] which can be seen from simulations.

Now, charge lost from the output cap per cycle is given by:

$$Q_{LOST} = C_{SW1} * V_{DD} + C_{SW2} * V_{DD} + |Q_I| + Q_{MISC} \quad [8]$$

where C_{SW1} is the cap of switch 1, C_{SW2} is the cap of switch 2, $|Q_I|$ is the magnitude of the inversion layer charge in the DTPA & Q_{MISC} accounts for other losses like leakage & the current needed for comparator (in the diode implementation).

Since, $|Q_I|$ forms a major portion of the Gate charge of the DTPA, we can write:

$$|Q_I| = \lambda * V_S * C_{GS}; \qquad \lambda \epsilon [0.8, 1] \text{typically} \qquad [9]$$

Now, the net charge transferred to the output cap is given by:

$$Q_{NET} = \mathrm{Q_{GAIN}} - Q_{LOST} = (\mu - \lambda) * \mathrm{V_S} * \mathrm{C_{GS}} - C_{SW1} * V_{DD} - C_{SW2} * V_{DD} - Q_{MISC} \qquad [10]$$

From the above eqn of Q_{NET}, it can be seen that the following three cases are possible:

- As mentioned above, max(μ)<min(λ), hence Q_{NET} is negative, & the output cap constantly loses charge.
- If the ranges are not satisfied & some how, μ>λ, then the other losses in the switches & the comparator can still make Q_{NET} negative overall, since V_S is of the order of few 100's of millivolts, whereas V_{DD} can be b/w 1V to 1.8V.
- Even if, Q_{NET} remains positive, it will be an extremely small fraction of V_S*C_{GS}. Much better performance can be obtained by just putting a passive capacitor instead of the DTPA since, the analysis of the passive capacitor won't involve the λ* V_S*C_{GS} term. Since, the other terms are not dependent on V_S*C_{GS}, they can be separately optimized in design & a higher positive value of Q_{NET} can be obtained, which is how this problem of losses did not occur up till now in the existing Energy Harvesting IC.

Hence, it was concluded that using a DTPA with the existing architecture doesn't offer any more benefit to the charging speed. Rather it just eats up the existing charge on the output capacitor during the boosting stage.

4. Maximum Power-Point Tracking in DC-DC converter

In this chapter, I will first explain the functioning of the current IC architecture and then in the subsequent chapters we shall move on to the design that I have implemented to improve the architecture further. Most of the content presented in this chapter draws from a paper by Gajendranath Chowdary & Shouribrata Chatterjee (4) which presents a detail design of a 300-nW, 50-nA DC-DC converter for energy harvesting applications. I will first explain the current technique used for extracting maximum power from the source & then in the subsequent sections, I will detail on how the current technique can be improved still. Note that I will not elaborate on the details of the design of the current circuit; rather my focus will be on the Maximum Power-Point Tracking (MPPT) Algorithm & on improving it by implementing a search algorithm through a digital control circuit.

4.1. Maximum Power-Point Tracking (MPPT)

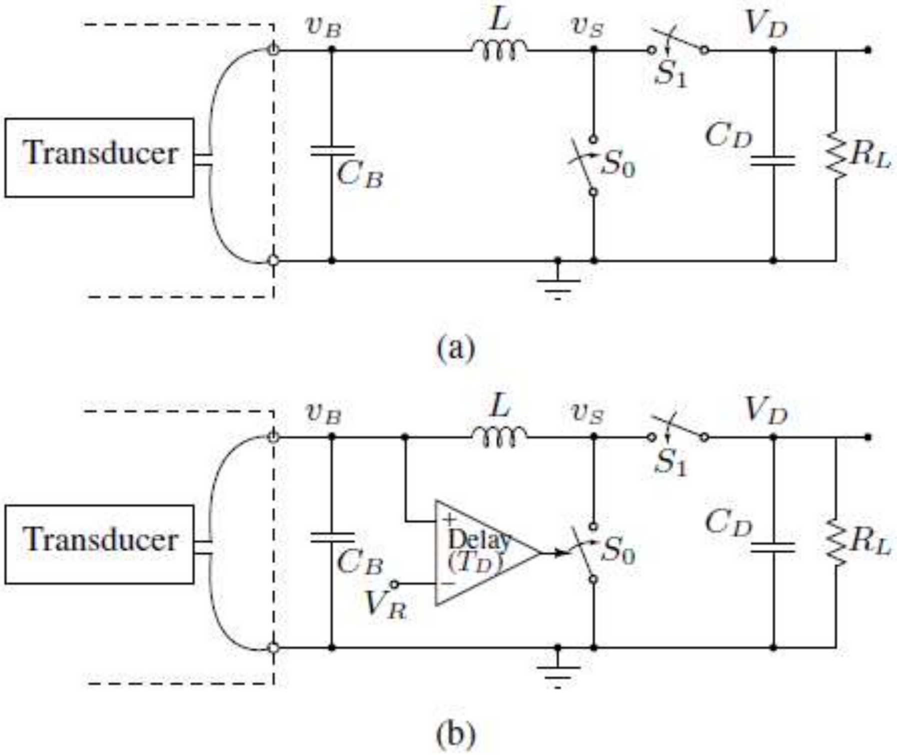

Figure 16: (a) Original DC-DC boost converter (b)Modified DC-DC boost converter (4)

Figure 16(a) above shows the original DC-DC boost converter that we discussed in chapter 1 of this thesis. This time we have changed the naming convention of elements in accordance with (4), but the circuit is essentially the same. We have the two switches

S_0 & S_1, an inductor L, an input capacitor C_B, and an output capacitor C_D. The input voltage can be boosted to larger output voltages by controlling the ON and OFF times of the switches S_0 and S_1 (4). First of all, we introduce some more naming conventions.

As was claimed earlier that the circuit in Figure 16(a) works in three phases, we name the phases as follows:

- **Wait**: When both switches S_0 & S_1 are open and only the bucket capacitor C_B is getting charged (4).
- **Energize**: When switch S_0 is closed & $S_1$is opened i.e. the inductor builds up current and is getting energized from the input voltage v_B (4). The rate of change of current through the inductor L is nominally v_B/L (4).
- **Dump**: When switch S_1 is closed & S_0 is opened i.e. the inductor dumps the built-up current onto the output capacitor C_D and this phase continues till the current in the inductor drops to zero (4). The rate of change of current through the inductor L is nominally V_D/L (4).

These three phases (quasi-)periodically repeat with a period of T_S, to harvest available power from the input (4). In energy harvesting applications, transformation ratios (V_D/v_B) can be as large as 20 (4).

The aim now is to obtain maximum power out of the above circuit. Apart from choosing the values of C_B & L appropriately, another factor that we can influence is the switching frequency of the switches S_0 & S_1. We now invoke the maximum power transfer theorem, according to which the effective input impedance of the DC-DC converter should be matched to the conjugate of the output impedance of the ambient-energy transducer in order to extract the maximum power out of the ambient-energy transducer. For the above circuit, this translates to maintaining v_B at a particular fraction of V_S (say 0.5) on average, in order to obtain maximum power transfer from the source to the DC-DC converter. For most kinds of sources & transducers, this fixed fraction is pre-determined and hence known in advance. For most sources, it is equal to 0.5, for some others like Photovoltaic cells it is close to 0.75.

So to regulate v_B at the required fraction of the source voltage, a slight modification is made. The switch S_0 is now controlled by a comparator output in the current architecture, as shown in Figure 16(b) above. The reference voltage of the comparator is set as the desired voltage to which v_B has to be regulated to. It can be shown that this approach tends to regulate v_B around the reference voltage $V_R$of comparator (4). Hence, the circuit tends to track the maximum power transfer point.

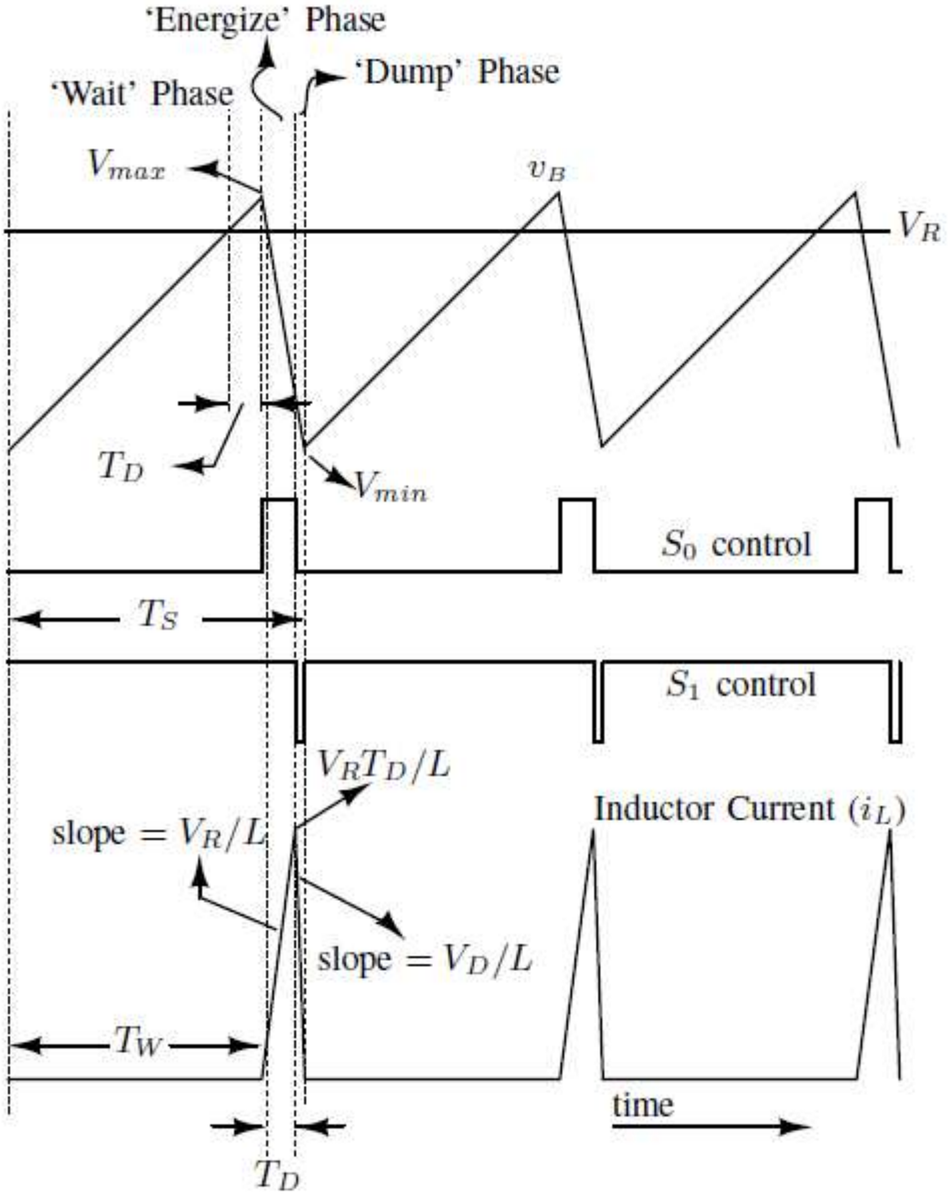

Figure 17: Waveforms showing vB, controls for switches S0 and S1, and current through inductor L. (S1 = PMOS switch) (4)

Figure 17 shows a timing diagram depicting the charging cycles & clearly illustrating the 3 phases: 'wait', 'energize' and 'dump'. The waveforms for v_B, control voltages of switches S_0 & $S_1 (PMOS\ switch)$ and the current i_L have been shown clearly.

Next, I shall quote some results & expressions derived in (4) directly without proof. First, it can be shown that a large value of the inductor L has to be chosen to minimize resistive losses & hence we cannot choose it to maximize the power transfer. The authors of (4) have chosen the optimum value for L as 47 μH. Seconly, the switching losses can be reduced if value of T_D i.e. the comparator delay is chosen to be large. But at the same time, it has been shown in (4) that the ripple voltage (v_{rp}) & the waiting time (T_W) increase as the comparator delay increases:

$$v_{rp} \approx \frac{V_R T_D^2}{2LC_B} \qquad (4)[11]$$

$$T_S \approx T_W \approx \frac{R_S T_D^2}{2L} \qquad (4)[12]$$

From the above equations, it can be seen that to control the ripple voltage & the time fraction of the 'wait' phase (and consequently the total time of a cycle) we need to keep the comparator delay T_D small. Hence, there is a trade-off involved. Also, note that the total time of one cycle is roughly the same as that of the 'wait' phase since the time taken by the 'energize' & 'dump' phases is typically negligible as compared to that of the 'wait' phase (4). But, as the switching period T_S is independent of the value of the cap C_B, we generally choose C_B to control the ripple voltage appropriately. Further, in (4) the optimum value of T_D has been chosen to minimize the total losses (resistive + switching).

4.2. Problems with the current architecture

It turns out that the above architecture has not considered a few points during calculations, which offer a further scope of improvement.

4.2.1. Long waiting time (T_W)

Since, neither of $T_D$or L were chosen to control the time taken by the 'wait' phase T_W, it turns out that both the 'wait' phase time & consequently the total charge cycle time is quite high. The only other factor that influences the 'wait' phase time is the source resistance R_S but that generally has a very high value of the orders of 10's of kilo-ohms to several mega-ohms and cannot be controlled by us.

To make some estimates, let us take some typical values for the parameters from (4). So we take the following values of the parameters from (4), $i_{L,max} = 5mA$, $L = 47\mu H, V_R = 50mV, V_D = 1.5\,V, T_D = 3.9\mu s\ and\ R_S = 10k\Omega$. Now, let us estimate the time for the three phases:

- $T_{wait} \approx \frac{R_S T_D^2}{2L} = \ 1.62ms$
- $T_{ener} \approx \frac{i_{L,max} * L}{v_B} = 2.5\mu s$
- $T_{dump} \approx \frac{i_{L,max} * L}{V_D} = 0.16\mu s$

It can be easily observed that the time for the 'wait' phase is roughly 650 times more than that of the 'energize' phase & about 10000 times more than that of the 'dump' phase.

Note that this implies that the whole apparatus including the inductor remains idle for most of the time during a cycle and hence can be shared for other tasks if needed.

4.2.2. High quiescent power losses

It should also be pointed out that in (4), only the resistive & the switching power losses have been considered for minimization. The power required to generate the control signals for S_0 & $S_1$and that required to operate the comparator has to be derived from the output capacitor itself & has to minimized separately.

It turns out that the comparator, which has to be running the whole time will burn out a lot of power. We can also note that the comparator is ideally needed only near the arrival of the 'energize' & 'dump' phases. It can be turned off during the wait phase, if such a mechanism could be developed which would allow us to detect the onset of the 'energize' phase some time in advance. This can save a lot of quiescent power losses since the comparator is the element which claims the largest power loss & the 'wait' phase typically takes about 99.8% of the cycle time.

4.3. Proposal for a new architecture

The above problem was thoroughly analyzed & several possible solutions were considered. I'll describe them briefly in this section.

4.3.1. Solutions Considered

- The first solution that was considered was to find a way to reduce T_W. Since, the 'wait' phase time depends only on R_S, T_D *and* L, we can see that touching any of the parameters cannot be a feasible solution since, as already pointed out, R_S is typically not under our control, $T_D$has been chosen to optimize resistive & switching power losses and L needs to have a large value to control the resistive power losses. So, we had to discard this approach.
- The second way could have been to detect the end of the 'wait' phase or the onset of the 'energize' phase somehow & then accordingly turn the comparator ON or OFF whenever needed. After very thorough brainstorming though, I realized that there is no feasible way to implement this, since all the information about the phases is generated only through the comparator. So the only way to detect this would be by using another comparator, which would totally defeat the purpose of putting the original comparator to sleep during the 'wait' phase. Hence. This idea also had to be discarded.

- Lastly, it was observed that the output of the comparator is almost periodic & hence, if we can replace it by a low-power tunable frequency oscillator that would lead to a huge cut in the power losses. This can be easily observed from the timing diagram in Figure 17. The only problem to sort out is to mimic the frequency of the comparator pulses through the oscillator and keep them synchronized if the comparator's frequency changes.

4.3.2. Proposed Solution

So finally it was proposed that the comparator should only be used initially. After v_B has stabilized, replicate the comparator's output pulse frequency with a variable frequency oscillator. The oscillator would generate a frequency proportional to a digital code entered into it.

To mimic the frequency of the comparator and to keep synchronizing with it regularly, once in a while (say after 1000 charging cycles), a Successive Approximation based Search Algorithm (SAR) can be used to find the ideal code, which generates the frequency closest to that of the comparator's output. The idea has been illustrated with a rough block-diagram sketch in Figure 18 below.

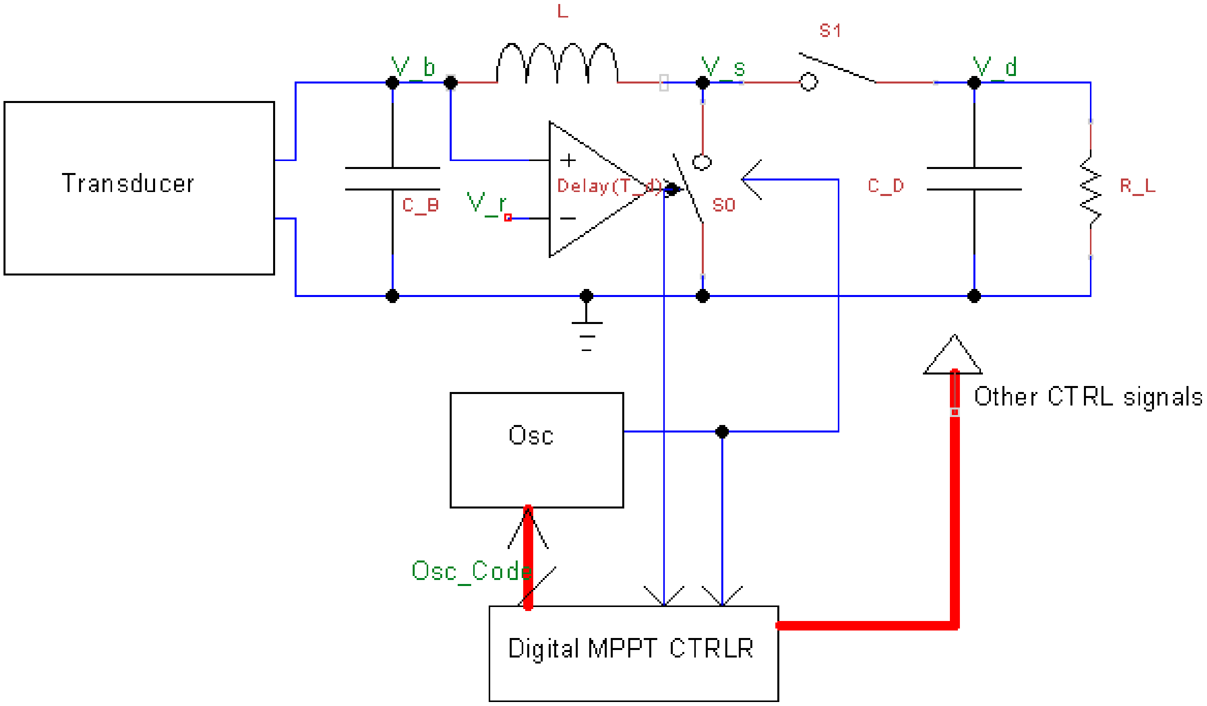

Figure 18: Proposed solution with a single source

Since, the variable frequency oscillator can be designed with a very low power requirement as compared to the comparator, this would lead to a huge amount of energy saving. Even though, this approach would require a separate digital controller to implement the SAR algorithm and the charging system, it is being proposed that it can be made still more efficient than using an analog comparator all the time.

Further, since the above solution only lowers the power requirements, but doesn't make more utilization of the waiting time, it was proposed that the system should be modified to share the inductor between multiple sources. Since, it has been observed that the inductor has lots of intermediate free time, it is viable to extract energy from multiple sources simultaneously while arbitrating carefully between their regular charging & SAR searching requirements, but at the same time sharing a lot of the hardware infrastructure and idle time amongst them to extract more energy in the same time.

Hence, the final proposal involved a design which consisted of the original energy harvester but with support for 3 sources: a PV cell, a Piezo source & an RF source which would all share the same inductor & a lot of the other existing hardware infrastructure. The second part of my project focuses on designing & implementing the digital controller which would act as both an arbitrator for the 3 sources and would implement all the decision making & sequential SAR searching algorithms necessary to operate the new energy harvesting architecture. An approximate block diagram of the final proposal has been shown in Figure 19.

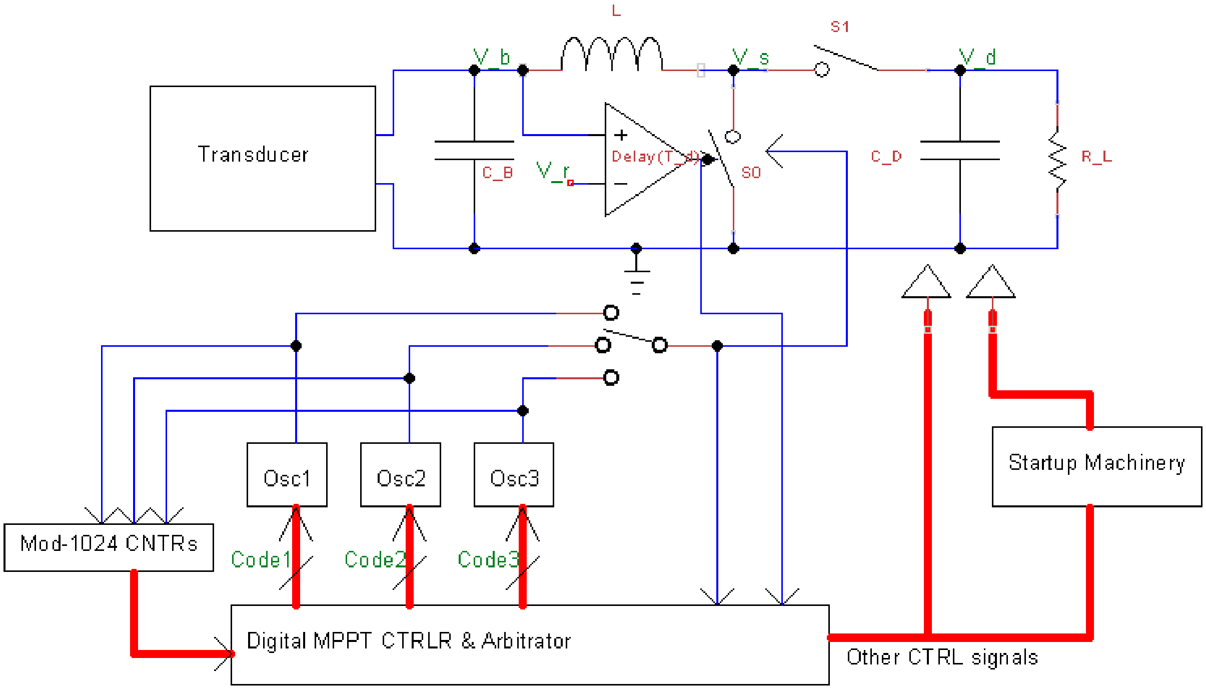

Figure 19: Proposed Solution with 3 sources

5. Digital Controller for a Single Source

I first started with the design of the digital controller for a single source, since it is always better to start with a small objective, get a small prototype running & then keep on making iterative improvements. So I'll start this chapter by elaborating on the design objectives for this controller with a single source. Later on, I present the design of the digital controller in the subsequent sections of this chapter and then it will be extended to 3 sources in the next chapter.

5.1. Design Objectives

- **Asynchronous design:** It must be noted that this controller won't be designed using a simple synchronous state machine design, since we do not have any global clock to synchronize with. Hence, the whole controller will be asynchronous and will run on triggers from the DC-DC boost converter.
- **Accuracy of SAR:** It needs to be taken care that our search algorithm converges to the correct frequency i.e. it converges to the discrete frequency in its working range closest to that of the comparator frequency.
- **Maintaining State of the System:** The controller must maintain a record of the correct activity that is being performed by the DC-DC boost converter. This information should be maintained in a register and will be helpful in tolerating errors & querying the state of the controller at any time.
- **Robustness to errors:** There is a lot of scope for errors in such a highly asynchronous design. For instance, requirement of the inductor for both the SAR search and the charge cycle simultaneously. In all such cases, the controller must ensure correct operation of the system and should be able to maintain a single state of operation, even if it involves bypassing some signals from the DC-DC boost converter. Also, the controller should keep running all the time and should not get stuck in any state.
- **Modularity:** The controller should be well-divided into appropriate independent modules. This will ensure an ease of debugging, maintenance and will also lead to proper segregation of the functional units in the controller block. Further, if the modules can be classified as shared or replicable, then it will also facilitate the job of extending the controller design for multiple sources.
- **Extensibility & Resource Sharing:** While the design progresses, it must be kept in mind at all times that the design is ultimately to be extended for multiple sources and hence, it should facilitate extensibility from the beginning itself. Modules must

be designed properly & segregated appropriately so that they can be shared amongst multiple sources with little or no effort spent in modifying them later.

- **Arbitration for multiple sources:** Though this feature is not as important at the moment as it will be when multiple sources are present, it still needs to be heeded. Even now, we have two operations i.e. SAR search and regular charging, which need to be arbitrated amongst themselves. Though, it will be easy to do it with only 2 operations, the method designed for implementing the arbitration procedure should be carefully chosen so that it can be extended later on for multiple sources with ease.
- **Responsibility towards load requirements:** It should also be noted that occasionally, but periodically there'll be a requirement to halt all ongoing operations & setup the boost converter to meet the load requirements. In such a case, it'll be needed to halt the controller's normal activities & suspend all operations temporarily till the load needs a power supply, after which the system should be able to resume its previous operations & continue working where it left off.
- **Diagnostic Outputs:** Finally, the controller will need to be tested later on, after the chip has been fabricated and it will be beneficial to add a few output pins which will bring out some signals which can help diagnose the system in case of failures. Such signals should be recognized & brought out as output pins.

5.2. SAR algorithm

SAR stands for Successive Approximation Register and the search algorithm is based on iterative scheme which starts of with an initial guess for the quantity to be searched & then successively improves the guess till it converges to an estimate closest to the actual value of the quantity to be searched.

For our purpose, the quantity to be searched for is the Comparator Frequency. We'll maintain a register with a binary code in it, which will govern the frequency of the Oscillator and we'll apply the SAR algorithm by successively changing this register code and try to converge to the closest possible estimate for the comparator frequency.

Let us define the comparator frequency to be f_C, the oscillator code to be $CODE$ & the maximum value of the code to be $CODE_{MAX}$. It is assumed that the oscillator frequency f_O is a monotonically decreasing function of $CODE$ and also that f_C lies in the range of frequencies spanned between $CODE = 0 \ to \ CODE = CODE_{MAX}$. Since, we're using a 9-bit register for storing the oscillator code, in our case the value $CODE_{MAX} = 2^9 - 1 = 511$. Finally in order to compare $f_C \ and \ f_O$, we keep a track of the rising edges of the oscillator's output waveform. We latch the rising edge on a flip-flop whose output shall be denoted by F_{out}. The flip-flop is cleared immediately after a comparison. The algorithm is as follows:

(1) Initialize $CODE = \frac{CODE_{MAX}}{2}$ $and\ SPAN = \frac{CODE_{MAX}}{4}$.

(2) Reset the oscillator & wait for the Comparator's rising edge.

(3) At the rising edge of comparator, check F_{out}.

(4) If $(F_{out} == 0)$ then this means that f_O is lesser than f_C and needs to be increased. Hence, do

$$CODE = CODE - SPAN$$
$$SPAN = \frac{SPAN}{2}$$

Else if $(F_{out} == 1)$, this means that f_O is greater than f_C and needs to be decreased. Hence, do

$$CODE = CODE + SPAN$$
$$SPAN = \frac{SPAN}{2}$$

(5) Clear F_{out} and reset the oscillator.

(6) If $(SPAN == 0)$, output $CODE$ else go to Step-(3).

The above algorithm follows a convergence rate linear in the number of bits in the CODE register. For a 9-bit register, the algorithm is guaranteed to converge in 8 steps. Figure 20 shows a graphical illustration of the convergence of the SAR algorithm with $CODE_{MAX} = D$.

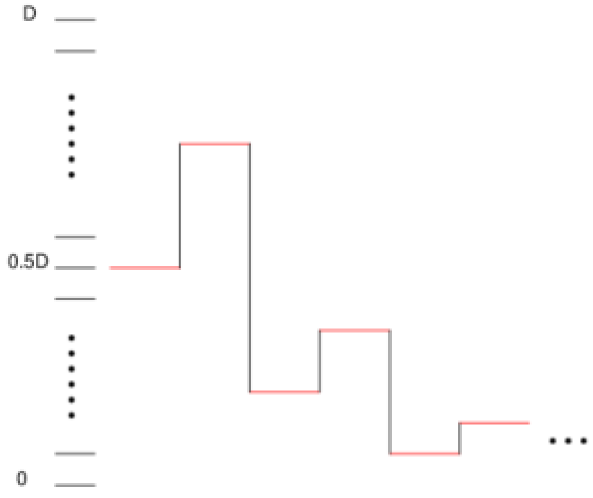

Figure 20: Illustration of the convergence of SAR algorithm.

5.3. Controller Design

In this section, I'll describe the final controller design that was implemented. But, I'll not go into logic gate or transistor level details, rather I'll focus on explaining the functionality of each module and how it meets the above specified design objectives. Appropriate circuit diagrams will be provided wherever they contribute to the understanding of the controller's operation.

5.3.1. Overall Controller Circuit Block

The overall block of the circuit is shown in Figure 21. The inputs and the outputs of the controller are clearly shown in the figure.

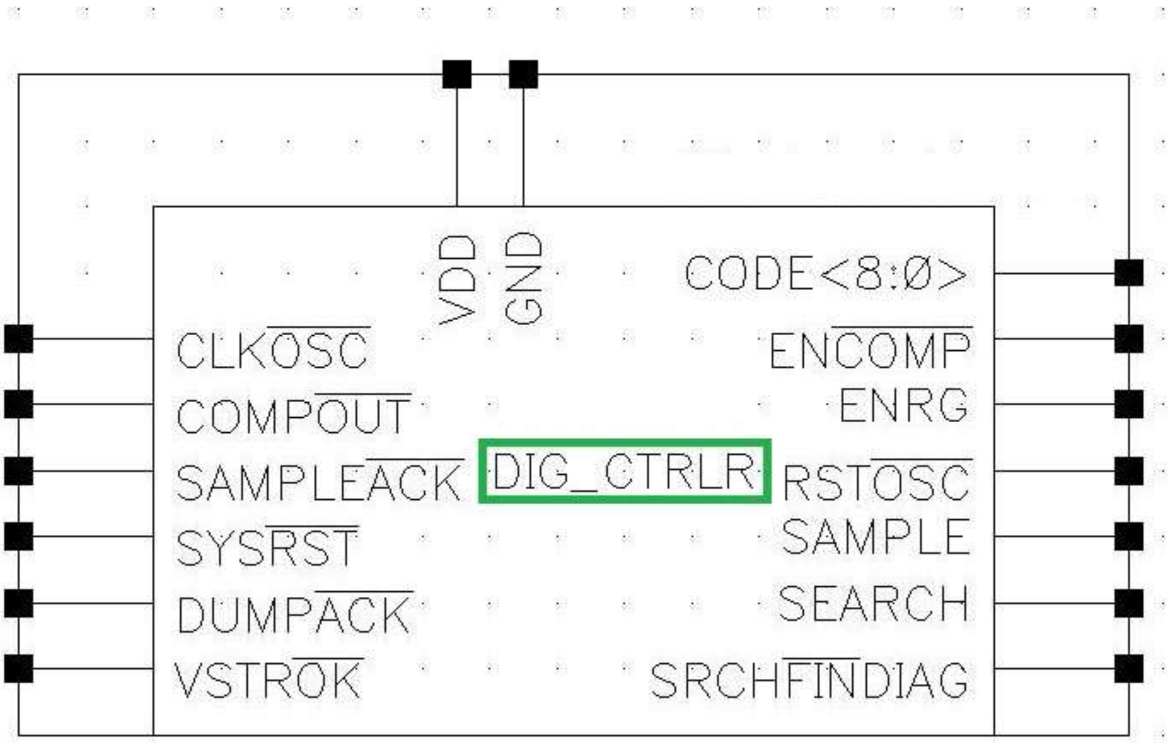

Figure 21: Symbolic block diagram of Digital Controller

I'll explain the inputs and the outputs in more detail.

(1) Outputs:-

a. CODE<8:0>: The 9 bit register which governs the oscillator frequency.
b. EN_COMP: The output signal which enables/disables the comparator for SAR.
c. ENRG: To signal boost converter to implement an energize-dump operation.
d. RST_OSC: Resets the oscillator to zero phase.
e. SAMPLE: To sample the comparator's reference voltage.

f. SEARCH: To tell oscillator that search has begun.

g. SEARCH_FIN_DIAG: Diagnostic signal indicating start/end of search operations.

(2) Inputs:-

a. VDD: Input Power pin

b. GND: Ground Pin

c. CLK_OSC: Input pin to take in Oscillator's output signal.

d. COMP_OUT: Input pin to take in comparator's output pulses.

e. SAMPLE_ACK: Input pin to acknowledge the finishing of a sampling operation.

f. DUMP_ACK: Input pin to acknowledge finishing of an energize-dump operation.

g. VSTR_OK: Indicates whether load needs to operate currently or not.

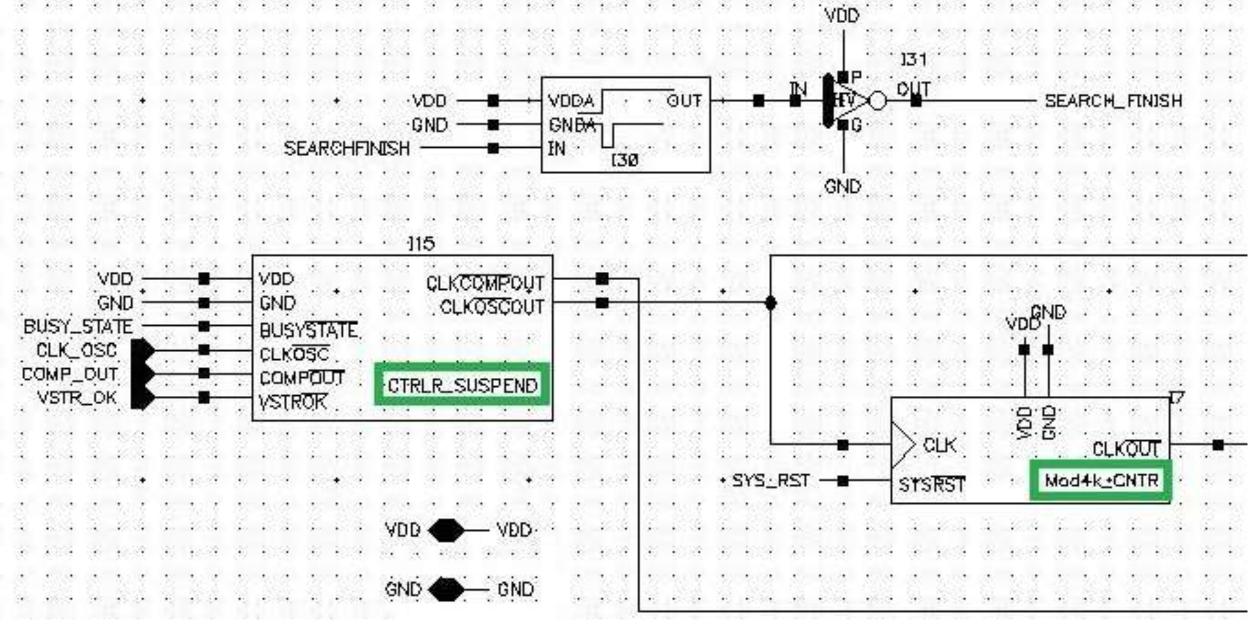

Figure 22: Inside the digital Controller, Part1

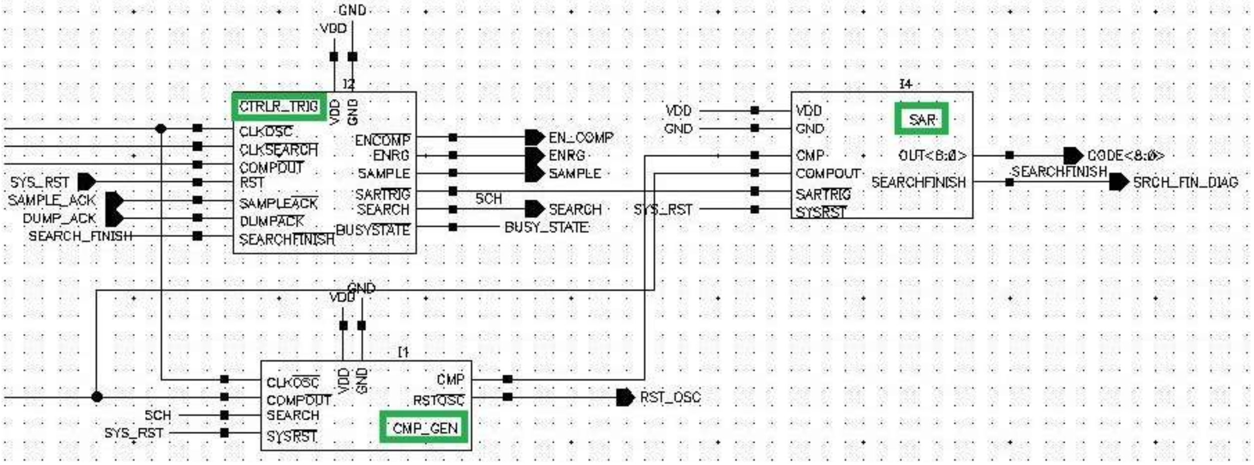

Figure 23: Inside the digital Controller, Part2

Internally the controller is divided into 5 major blocks as shown in Figure 22 and 23 above. The 5 blocks are as follows:

- CTRLR_SUSPEND
- Mod4k_CNTR
- CTRLR_TRIG
- CMP_GEN
- SAR

In the following sub-sections I'll describe the functioning of these internal blocks without going into very tedious logic gate level details. Though, wherever it is feasible I'll also provide the logic circuit of the module to facilitate understanding, if the reader is curious.

5.3.2. CTRLR_SUSPEND module

This module takes care of the load requirements i.e. it takes as its input the signal VSTR_OK. Whenever the signal goes high, this block starts checking to see if the controller is busy or not. While the controller is busy, the module does nothing. As soon as the controller becomes idle it blocks both the oscillator & comparator outputs to the controller due to which no further actions can be taken. Hence, it effectively suspends the controller till the load is active. After the VSTR_OK signal goes down, the clocks are released back to normal again. The logic circuit for the module is as shown below in Figure 24.

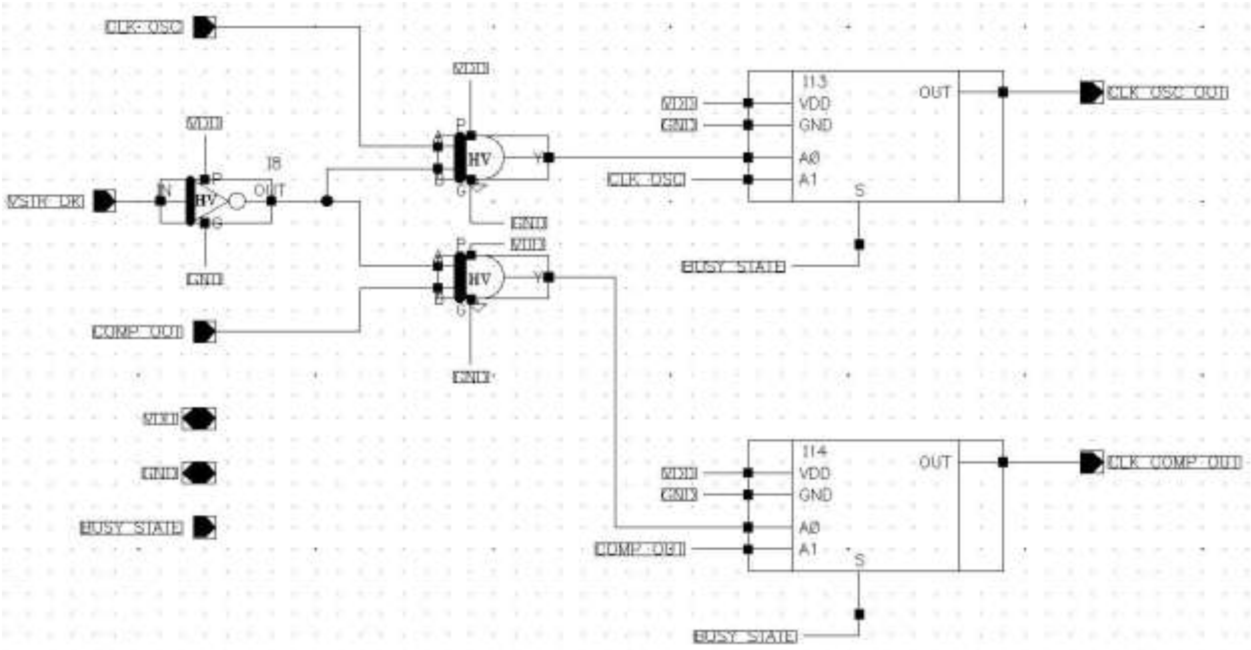

Figure 24: Logic Circuit for CTRLR_SUSPEND module

5.3.3. Mod4k_CNTR module

This module is just a standard mod-4096 asynchronous counter comprising of 12 T-flipflops stacked together in the clock-divider configuration.

5.3.4. CMP_GEN module

This module generates the result of comparison b/w the comparator frequency & the oscillator frequency which is needed by the SAR search module. It simply latches a HIGH on the rising edge of the oscillator clock, while waiting for the rising edge of the comparator clock. At the rising edge of the comparator clock, it checks the latched bit to see, if it is a 0 or a 1 and sends its output to the SAR block accordingly, after which it resets the oscillator. A logic circuit for this module has been shown in Figure 25.

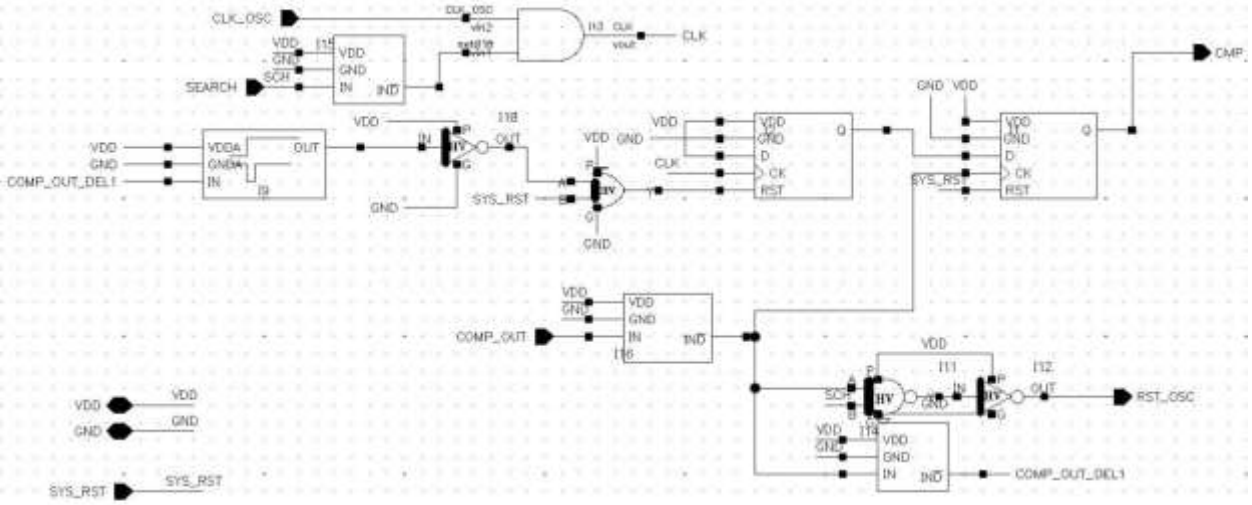

Figure 25: Logic Circuit for CMP_GEN module

5.3.5. SAR module

This module implements the actual search algorithm that was described in section 5.2 of this thesis. A logic diagram of the implementation has been shown in Figure 26. This circuit contains 9 registers in a shift register configuration SR<8:0> which maintain the current position to be searched. Another set of registers with outputs ACC<8:0> are the registers which contain the current code being searched.

The module is initialized at the SAR_TRIG pulse from the CTRLR_TRIG module or the SYS_RST pulse from outside the controller. The SR shift register is initialized with “100000000” which then shifts right on every iteration in order to maintain the current position being searched in the SAR algorithm. Note that this can be done, since a division by 2 is equivalent to a right shift. After the algorithm has been understood, the implementation shown in Figure 26 is pretty straight-forward to understand is hence left to the reader’s understanding.

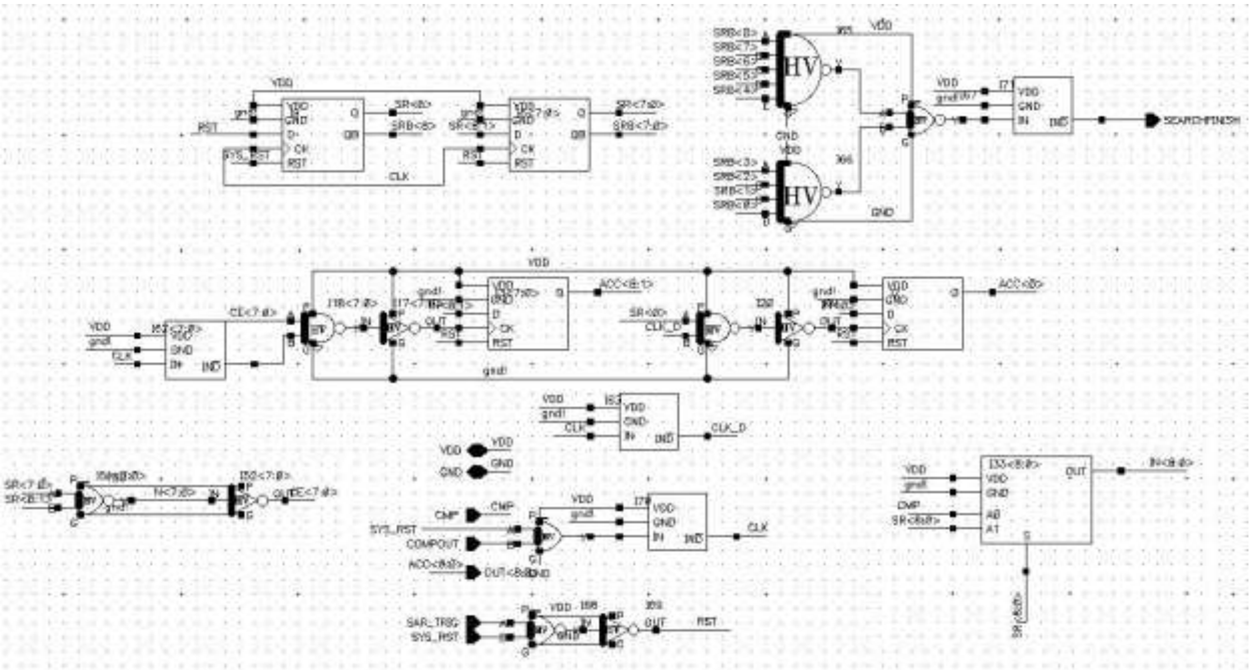

Figure 26: Logic circuit of SAR module

5.3.6. CTRLR_TRIG module

This is the module which performs a variety of triggering functions. First of all, it maintains a record of the current state of the controller i.e. it keeps track of the current operation being performed by the controller & the Energy Harvesting circuit. This makes it robust to errors since any new incoming operation is only approved after ensuring that the controller is not already busy with some previous operation. In case the controller is busy, the incoming operation is queued and remains so till the controller becomes idle again.

This controller keeps track of all the incoming charge & search requests and maintains a priority order for them. Typically, the way this module has been implemented, search requests are given a higher priority over charge requests, since search operations are rather infrequent and if they are preempted by charge requests, then they may remain in the queue forever.

The module triggers an energize-dump command (ENRG) or a search command (SAR_TRIG) as its output whenever needed. It also resets these requests as well as the state of the controller appropriately whenever a previously ongoing operation terminates.

Since this module performs a lot of operations its logic diagram is too big to present in a single piece hence, it is being presented in small pieces. Figure 27 shows the logic circuits of the registers which keep track of the state of the controller, as well as those which latch the incoming requests for energize-dump or search.

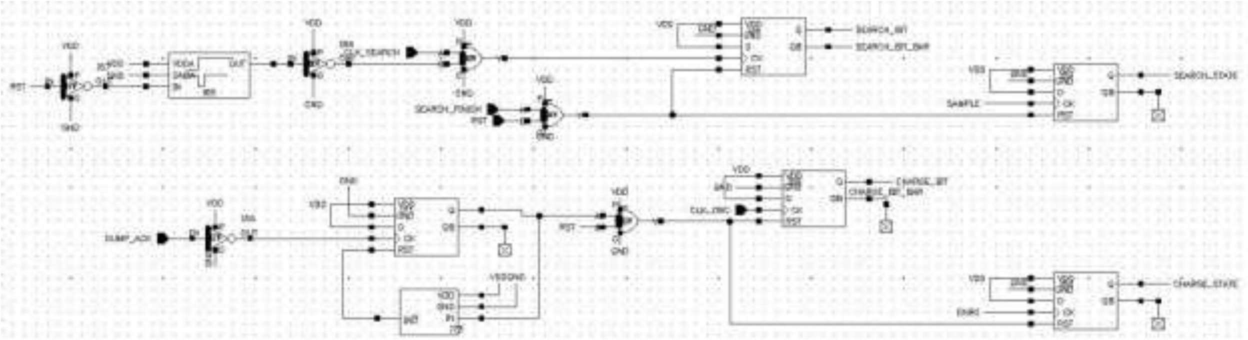

Figure 27: State of the controller & bits of the request queue.

Figure 28 shows the generation of the BUSY_STATE signal and its delayed complement which are checked everywhere throughout the circuit in order to avoid any kinds of clashes between various requests.

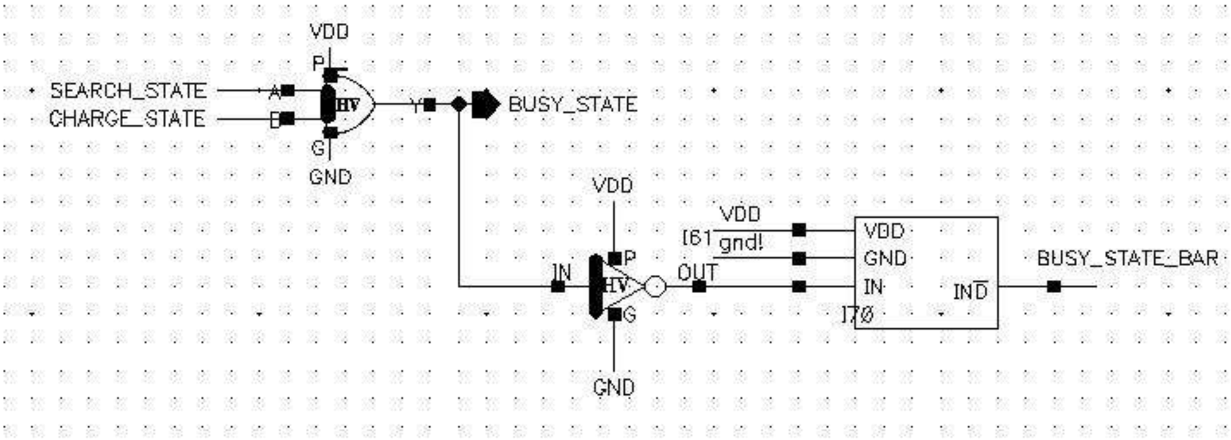

Figure 28: BUSY_STATE signal generation

Figure 29 shows the circuit which generates the ENRG signal to trigger an energize-dump operation and the circuit which generates the SAMPLE signal which is needed to sample the reference voltage off the comparator before starting a search operation.

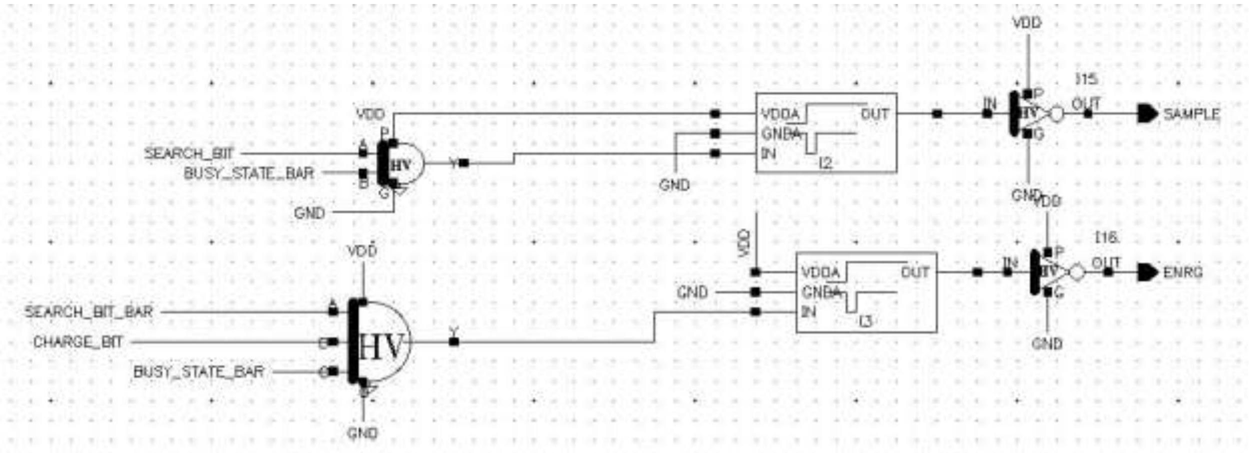

Figure 29: SAMPLE & ENRG operations

Figure 30 shows the generation of the SAR_TRIG signal which triggers a search operation. For the search operation the first three comparator pulses are skipped to give some time for the comparator pulses to settle down, once the EN_COMP signal goes up. Only after that is the SAR_TRIG signal is turned on.

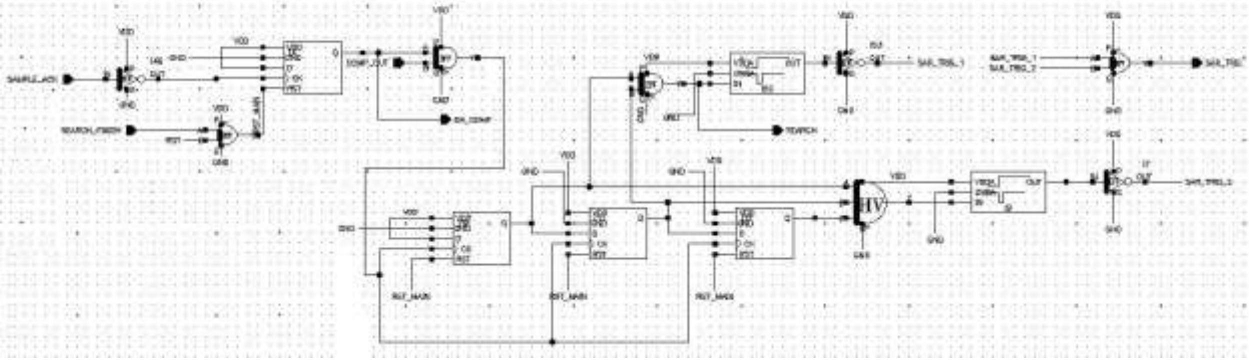

Figure 30: Generating the SAR_TRIG signal

This completes the design of the controller for a single source. In the next section we present a small simulation to demonstrate the effect of the SAR algorithm.

5.4. Simulating the SAR search algorithm

In this section, a simulation of the SAR search algorithm has been presented. We first prepared a test bench containing the complete digital controller & a variable frequency oscillator. Several voltage sources were included to give power and simulate the presence of the comparator. Finally some delay blocks were kept to emulate the effects of 'sampling' & 'energize-dump' and give back the acknowledgement pulses. The schematic of the test bench has been shown in Figure 31.

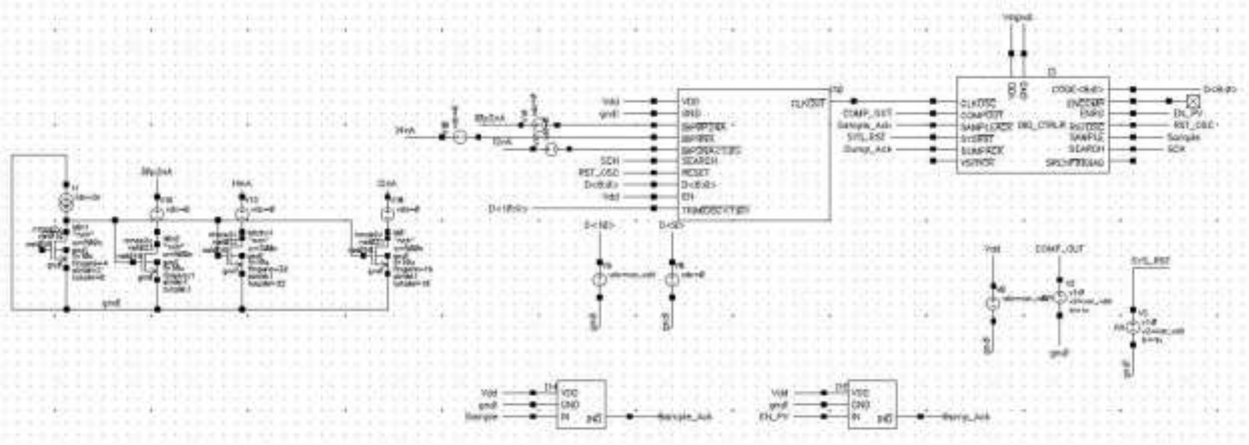

Figure 31: Test Bench schematic

Figure 32 shows small 2.5 ms run of the simulation. The curve on the top shows the pulses from the comparator i.e. the COMP_OUT signal. It has been given with a time period of 75us. The bottom signal is that of the oscillator output i.e. OSC_OUT.

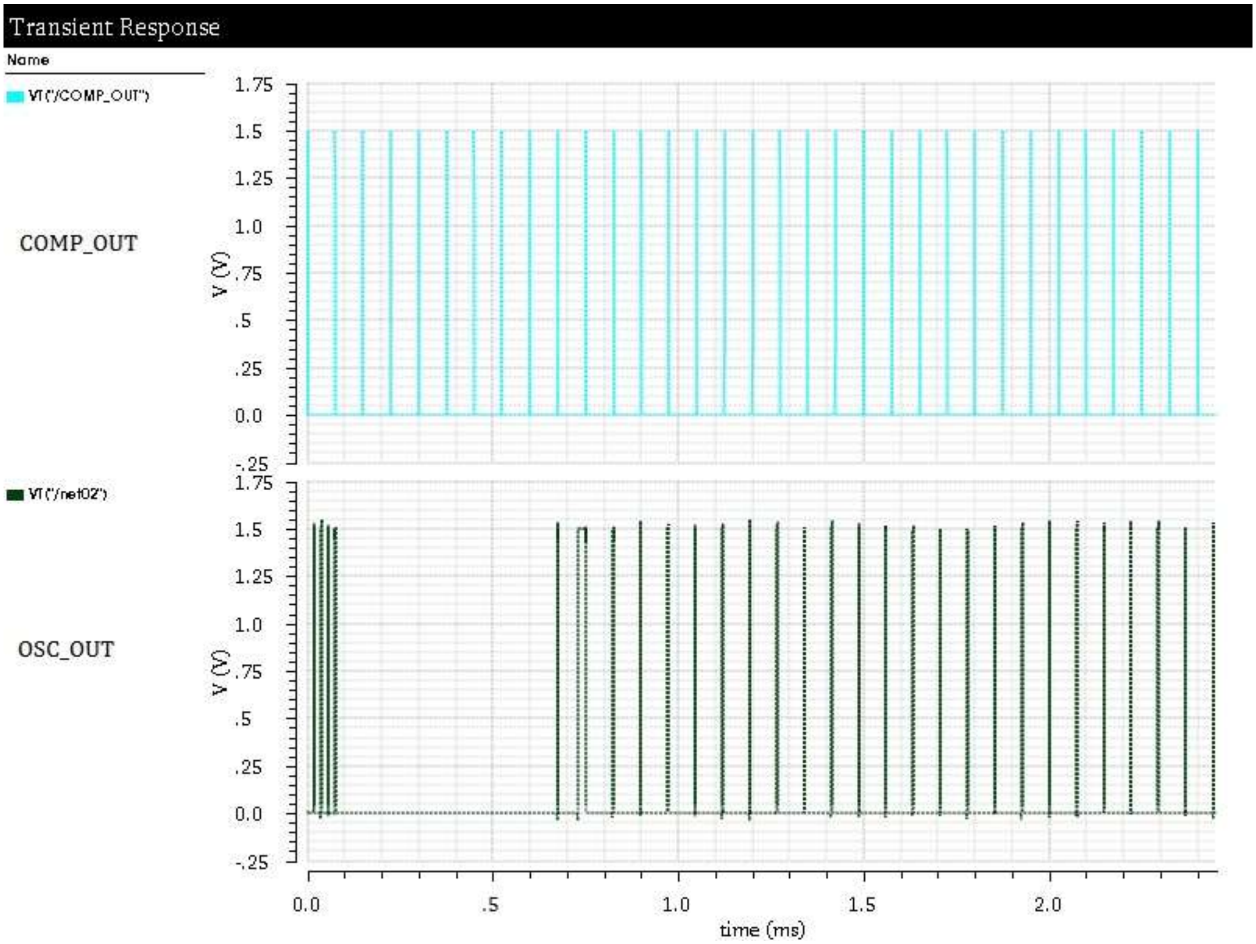

Figure 32: Demonstration of the SAR algorithm

It can be observed that it begins with a very high default frequency and then the signal disappears for some duration of time around 0.1ms to 0.7ms. This is roughly the time when the SAR algorithm is trying to match the frequencies of the comparator & the oscillator outputs. The search continues till about 0.8ms after which it can be seen that the comparator & the oscillator output are pretty much in sync with each other, both in frequency and in phase. It was observed that the final time period of the oscillator output was 75.6us which is quite close to that of the comparator output.

6. Extension to Multiple Sources

In this chapter we shall extend the controller that we designed in the previous chapter to three sources i.e. Photovoltaic (PV), Piezo (PZ) & RF. Since the previous design was made keeping the extension & the arbitration requirements in mind, it is fairly simple to extend the design for multiple sources. We shall quickly demonstrate the extension for three sources in the sub-sections of this chapter, providing adequate explanations wherever they are necessary.

6.1. Overall controller block

The symbol for the digital controller is pretty much the same except that many of the input and output signals this time have three copies, one for each of the sources. Figure 33 shows the symbol block of the new controller.

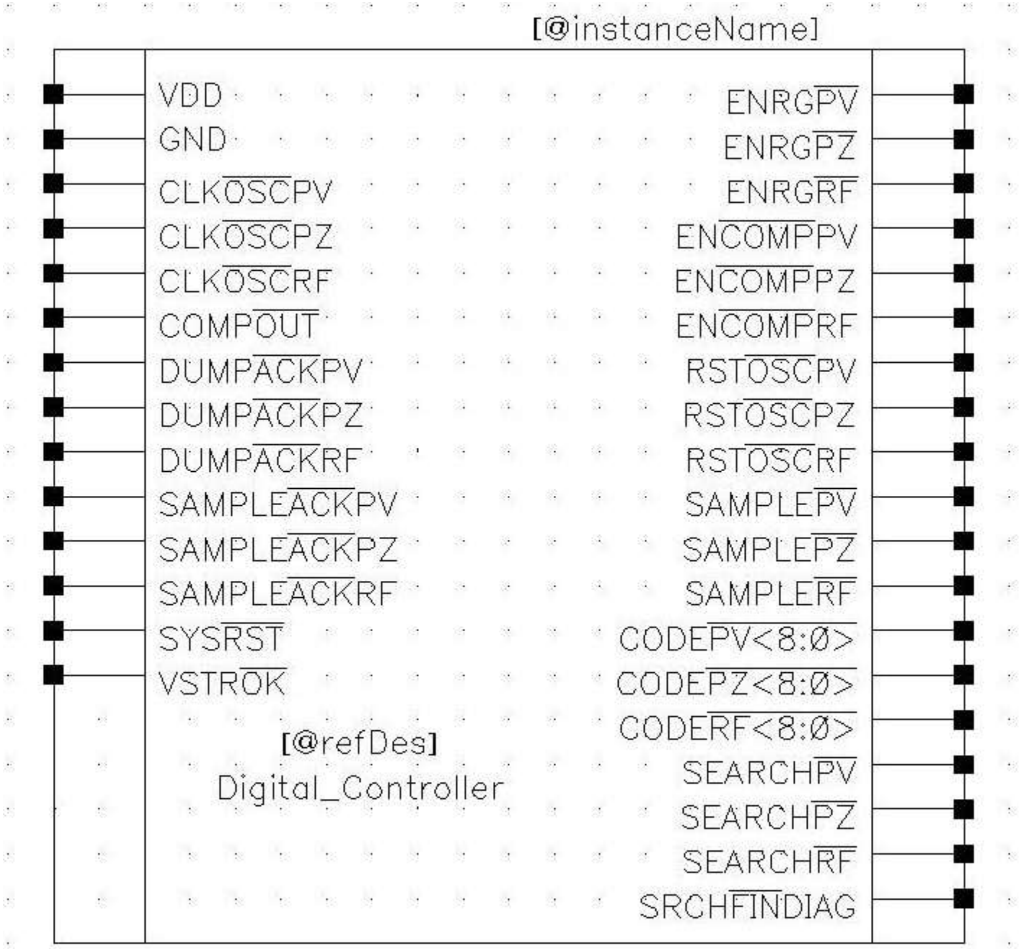

Figure 33: Symbol of the new controller

The internal structure of the controller is shown in Figure 34.

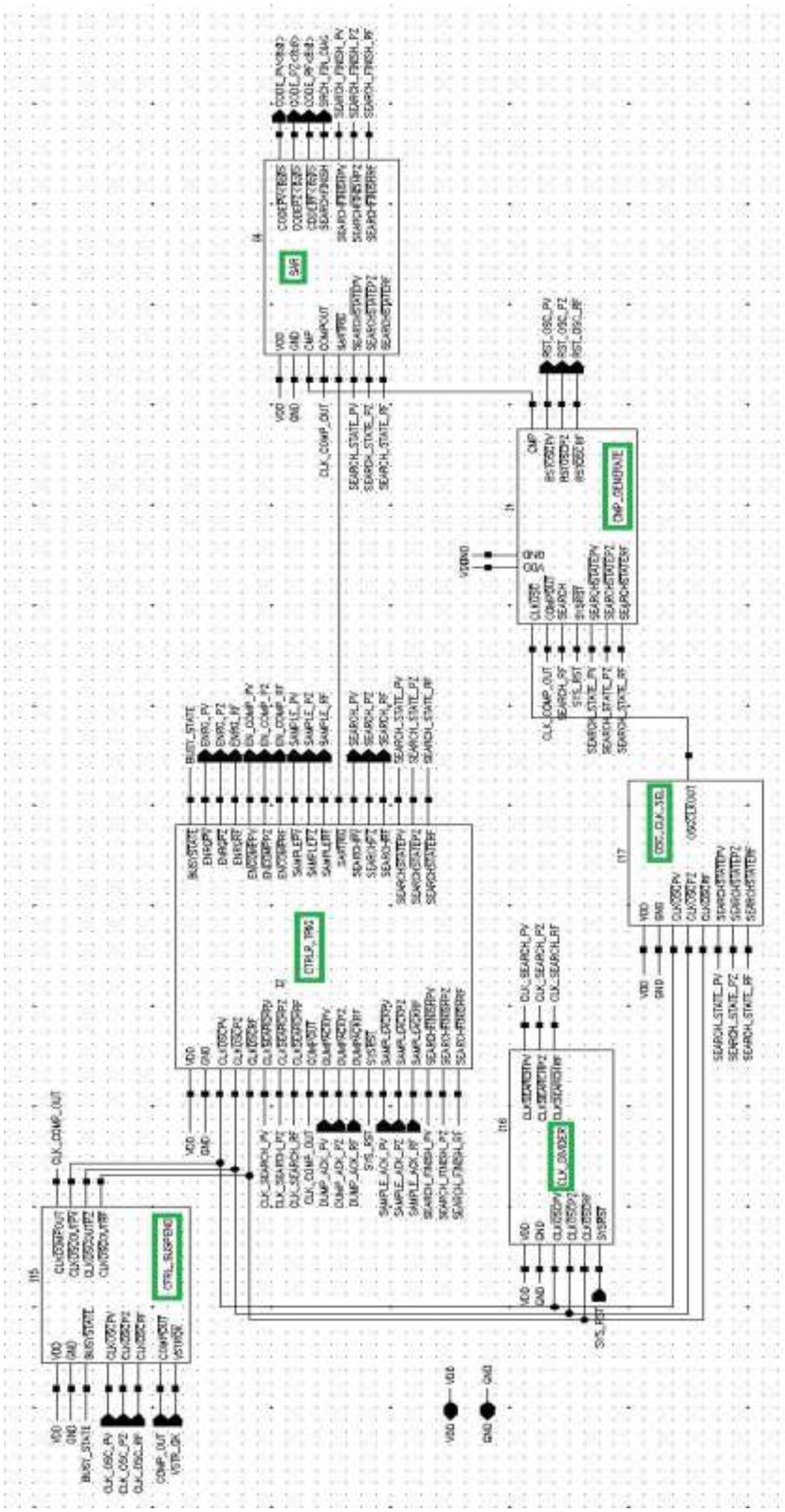

Figure 34: Internal block diagram of the Controller

Once again we can identify the same major blocks inside the controller, except there has been an addition of one extra block. These internal blocks are as follows:

- CTRLR_SUSPEND
- CLK_DIVIDER
- CTRLR_TRIG
- OSC_CLK_SEL
- CMP_GEN
- SAR

Most of the above blocks have changed very little in terms of functionality except that they have multiple copies of an input or output, one for each of the three sources.

A detailed description of the changes and of the new block is presented in the following sections of this chapter.

6.2. CTRLR_SUSPEND module

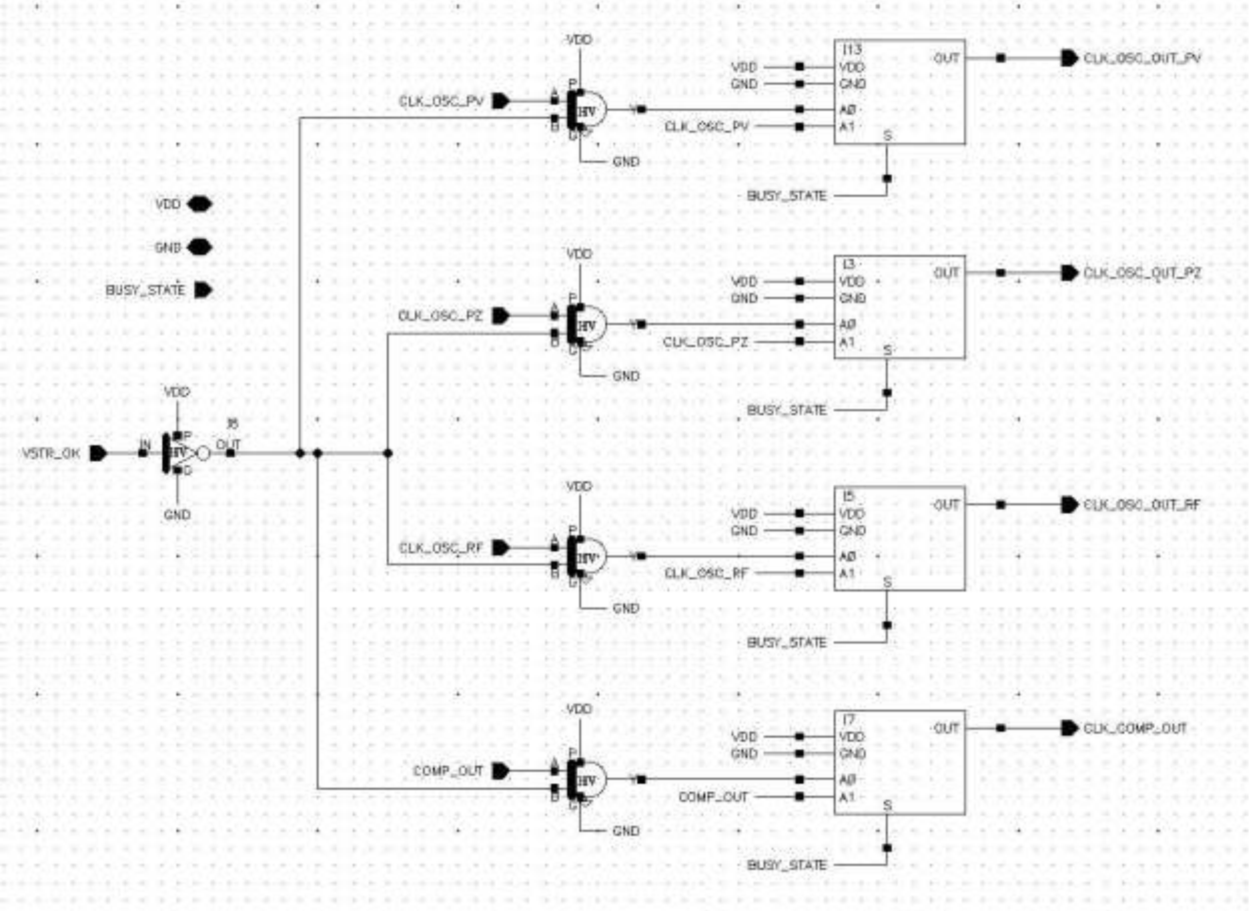

Figure 35: Logic circuit of new CTRLR_SUSPEND module

Figure 35 shows the logic circuit of the new CTRLR_SUSPEND module. As can be easily observed here, there is not much of a change in the logic except that this time, there are three oscillator clocks to be suspended and resumed as compared to a single oscillator clock last time. Hence the clock has four output signals instead of the previous two.

6.3. CLK_DIVIDER module

This module just replaces the previous mod-4096 counter. It is a simple extension containing three such counters and hence it generates the search clocks required for the three sources by dividing their charging pulses by 4096.

6.4. OSC_CLK_SEL module

This is a module which was not already present in the previous controller. It is required since the SAR & the CMP_GEN module still work for a single source at a time. Hence, this module simply checks the state of the control in order to send the correct oscillator clock to the CMP_GEN module for comparison with the comparator clock on every iteration. The logic circuit is shown in figure 36.

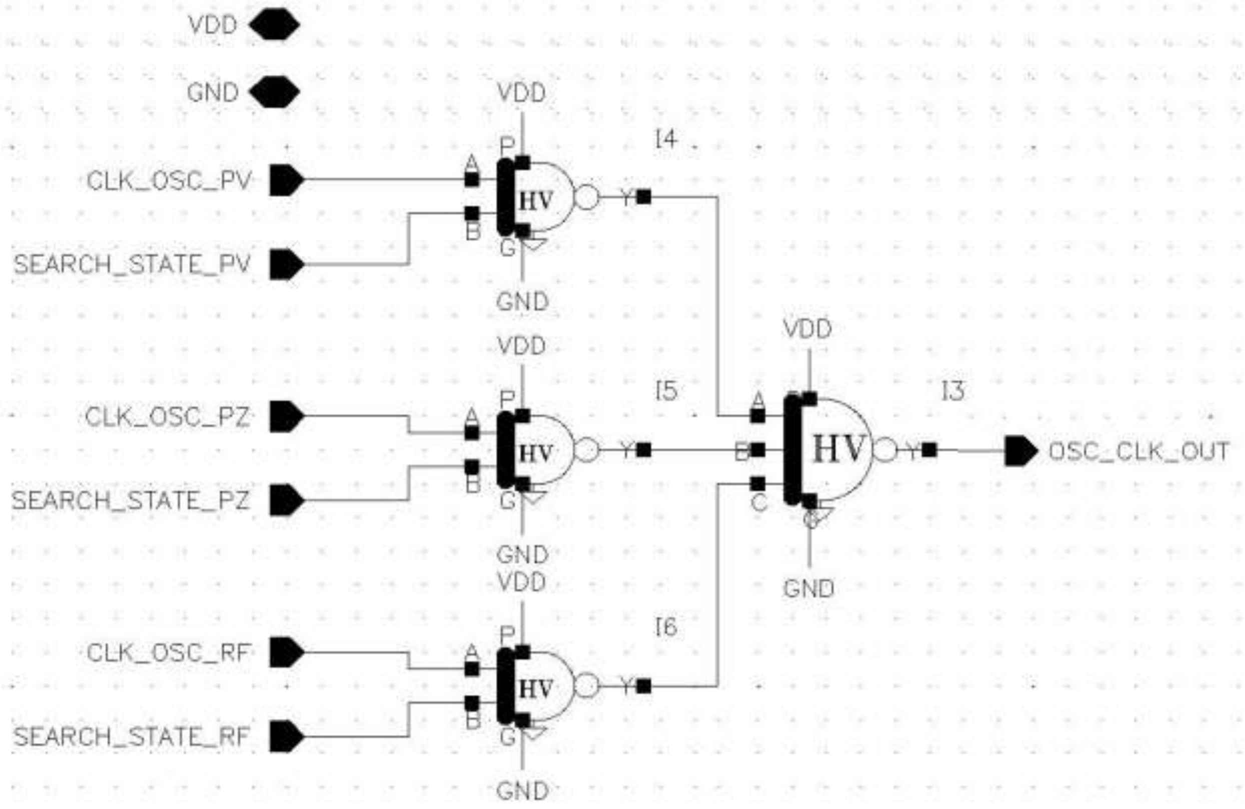

Figure 36: Logic circuit of OSC_CLK_SEL module

6.5. CMP_GEN module

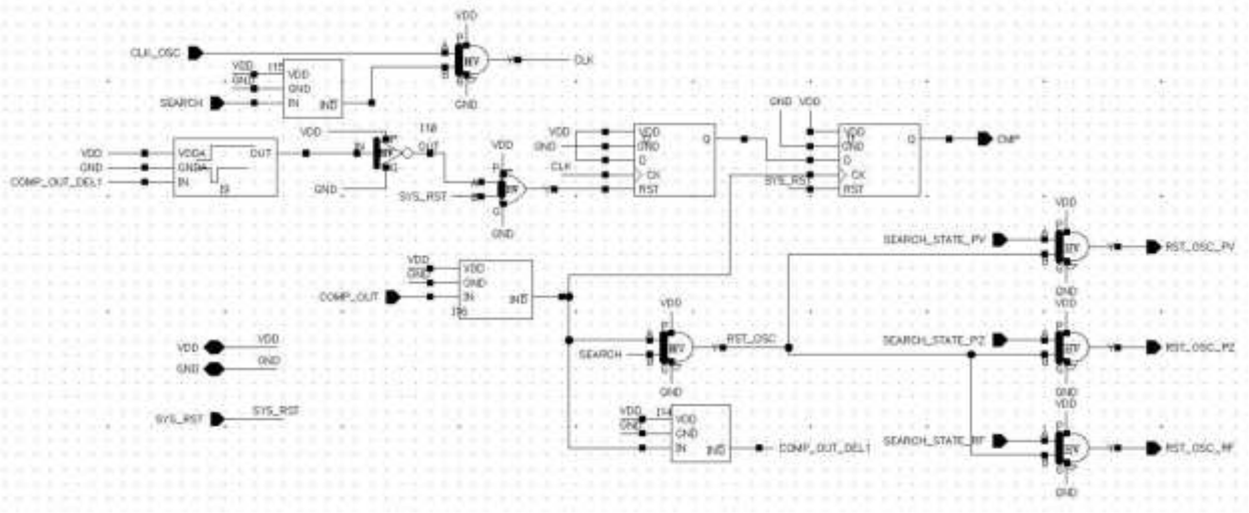

Figure 37: Logic circuit of CMP_GEN module

Figure 37 illustrates the logic circuit of the CMP_GEN module. Its function hasn't changed in this new controller, nor has its circuit diagram changed much, except for the fact that its output RST_OSC has been replaced by three counterparts & hence, it needs to reset the right oscillator while searching.

6.6. SAR module

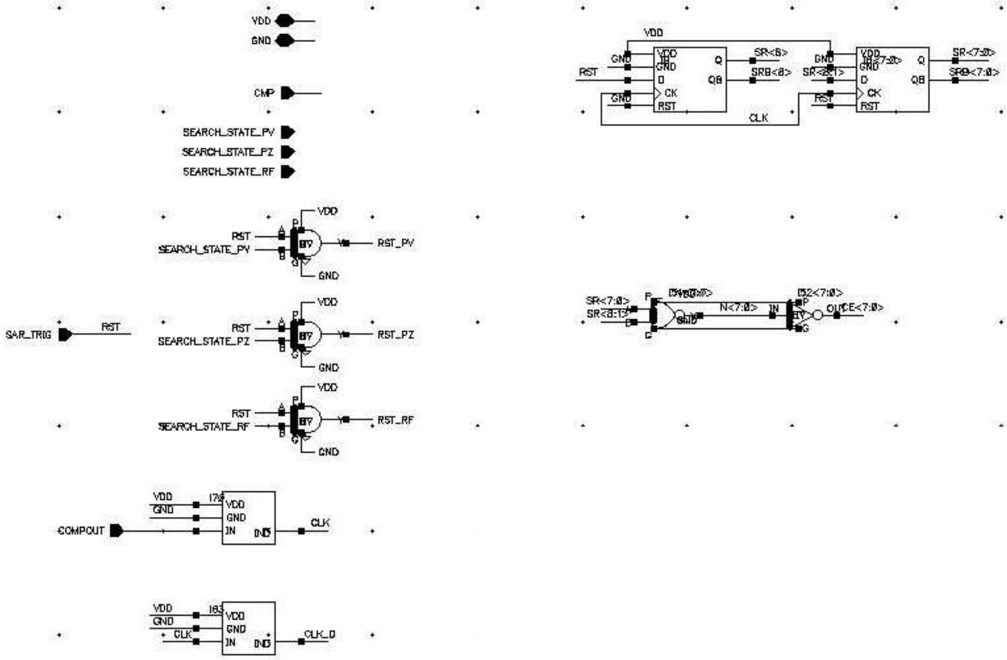

Figure 38: Logic circuit of SAR module, Part1

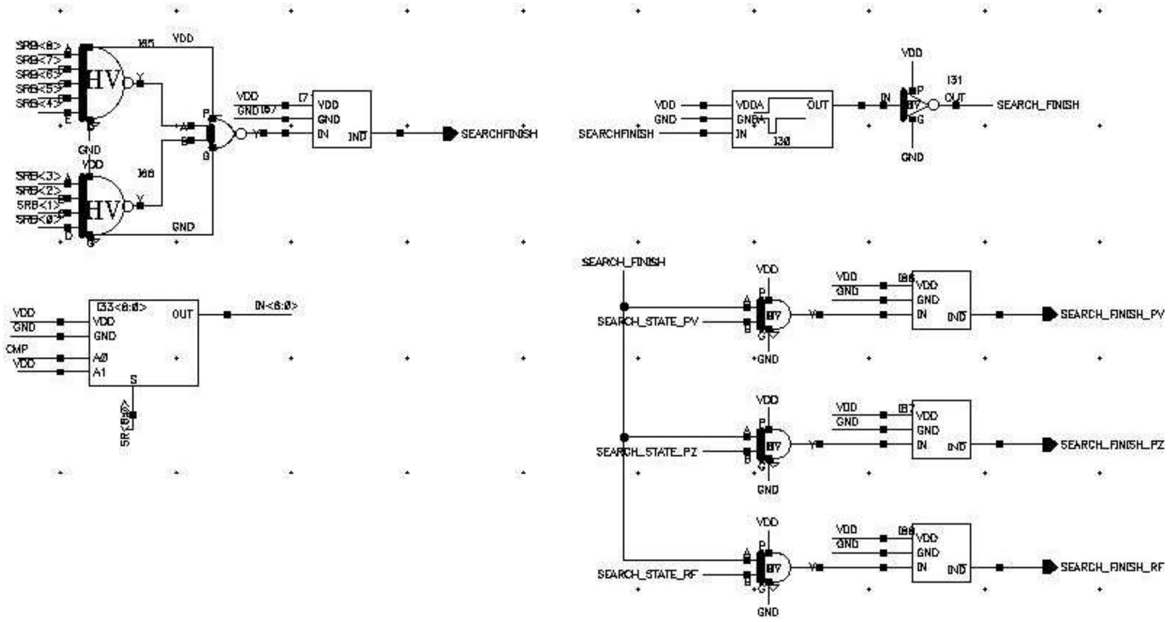

Figure 39: Logic circuit of SAR module, Part2

Figure 38 & 39 show a part of the logic circuit of the SAR module. As can be seen here, the status registers are still intact since they'll be shared amongst all the three sources owing to the fact that only a single source can undergo the SAR search at a time. Some other signals like SEARCH_FINISH & RST have had duplicate counterparts added though, in order to support multiple sources.

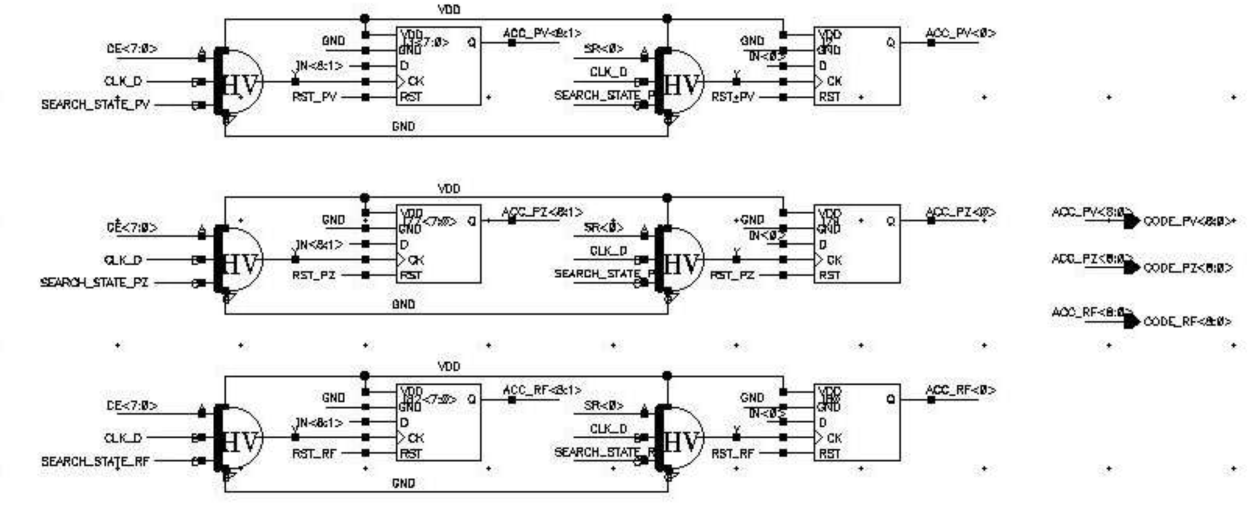

Figure 40: Logic circuit of SAR module, Part3

Figure 40 shows the registers which held the code for the oscillators. Obviously, three copies of each register are now needed to run the SAR search. But, during any search operation only one of the registers participates in the search while the other two hold off their codes intact.

6.7. CTRLR_TRIG module

Once again the function of the module remains same i.e. to maintain the state of the controller, handle all the search & energize-dump requests, arbitrate amongst them and trigger the right operations at the right time. Only this time, the arbitration has extended to 6 operations and hence requires more checks than previously. The priority order during arbitration has been extended as follows:

$$SRCH_{PV} > SRCH_{PZ} > SRCH_{RF} > CHARGE_{PV} > CHARGE_{PZ} > CHARGE_{RF}$$

Search operations have been given higher priority than charge operations to prevent them from suffering from starvation in the priority queue, sincere the charge operations are very frequent.

At most places, the only change has been the creation of two duplicate copies of certain hardware in order to serve three sources instead of one. The sub-parts of the circuit are illustrated in figures 41 to 47 below:

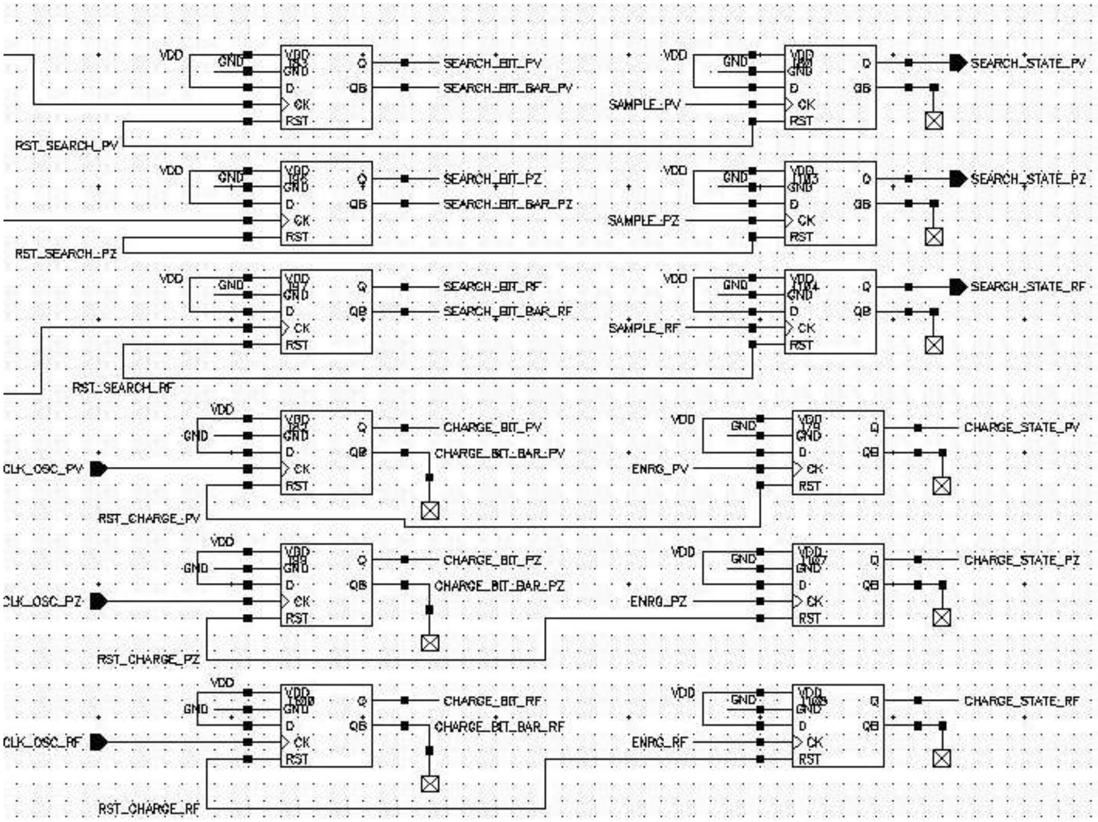

Figure 41: Logic circuit showing the State maintaining mechanism & the Search & charge request registers

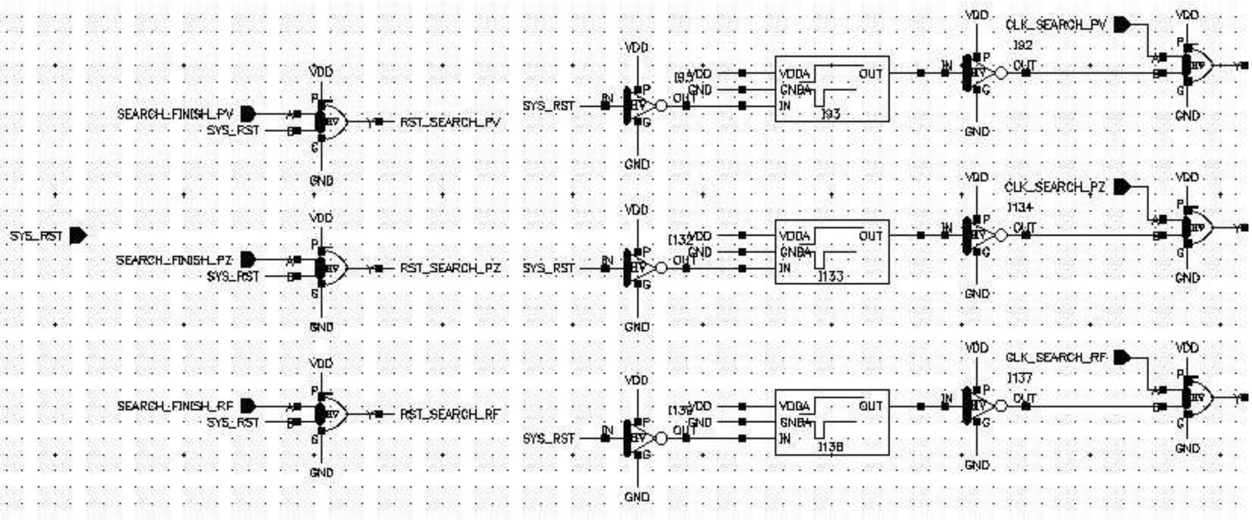

Figure 42: Logic circuit showing the reset and clock mechanism for state maintaining registers

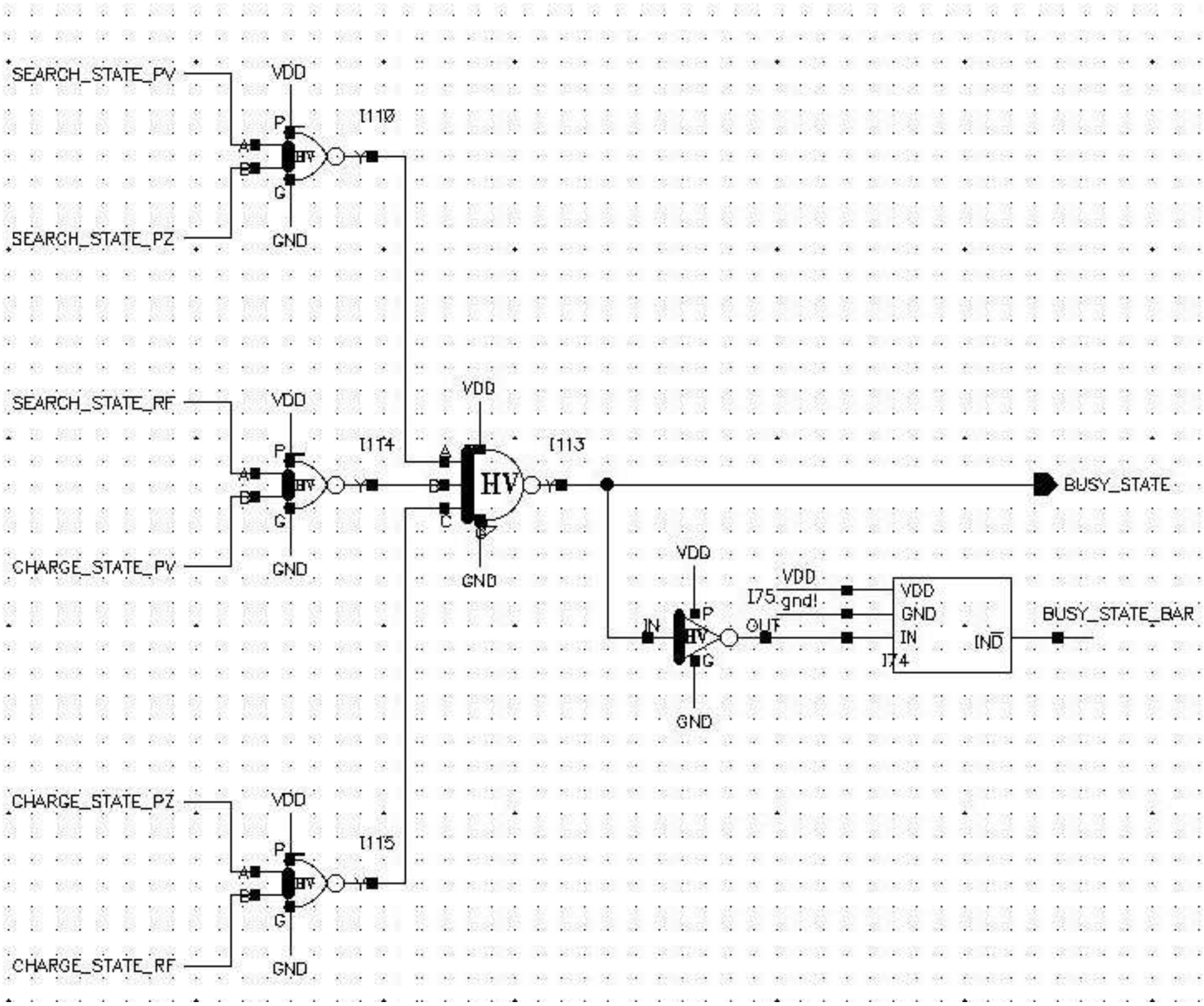

Figure 43: Generation of BUSY_STATE signal

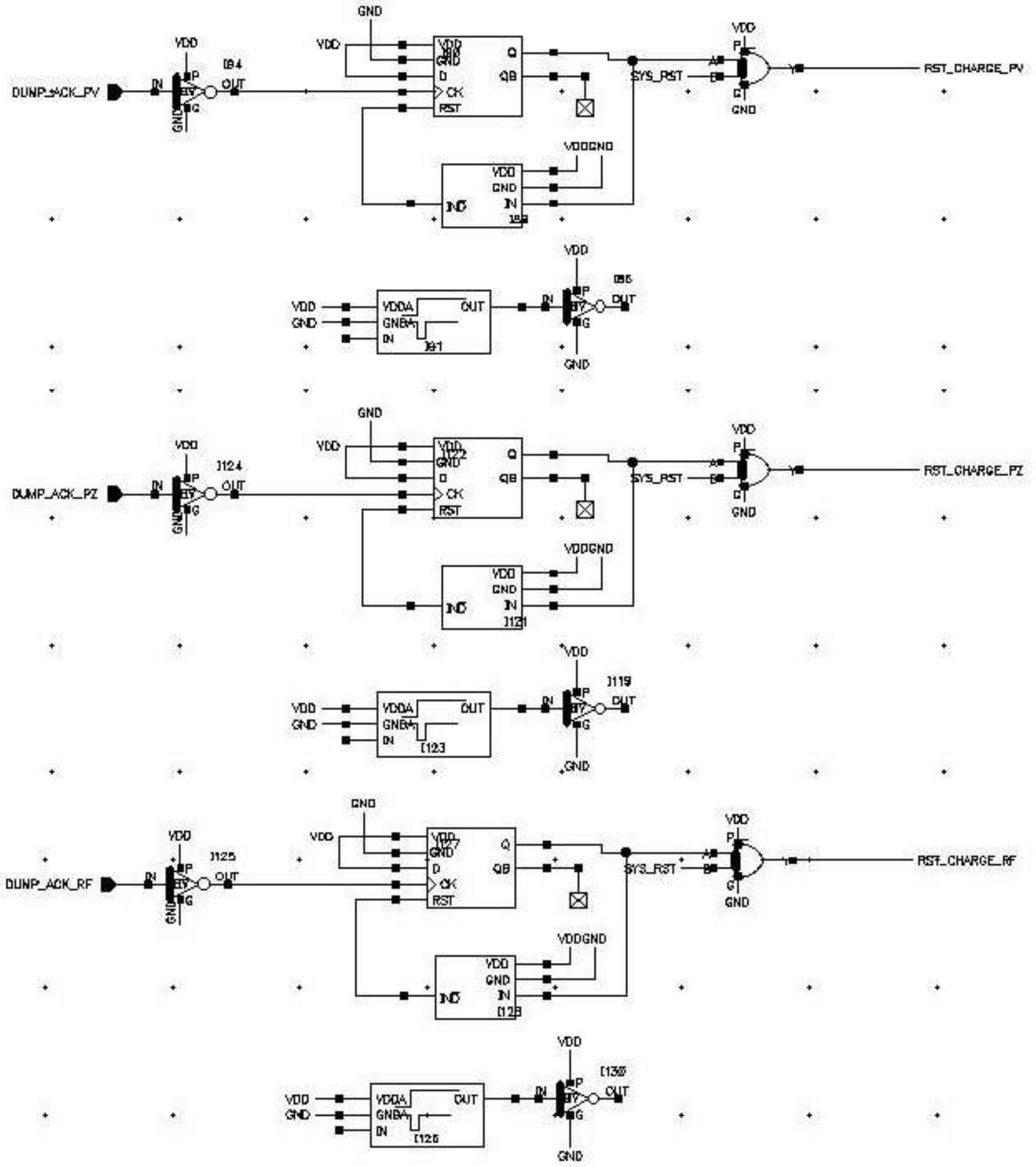

Figure 44: Resetting mechanism for Charge request registers

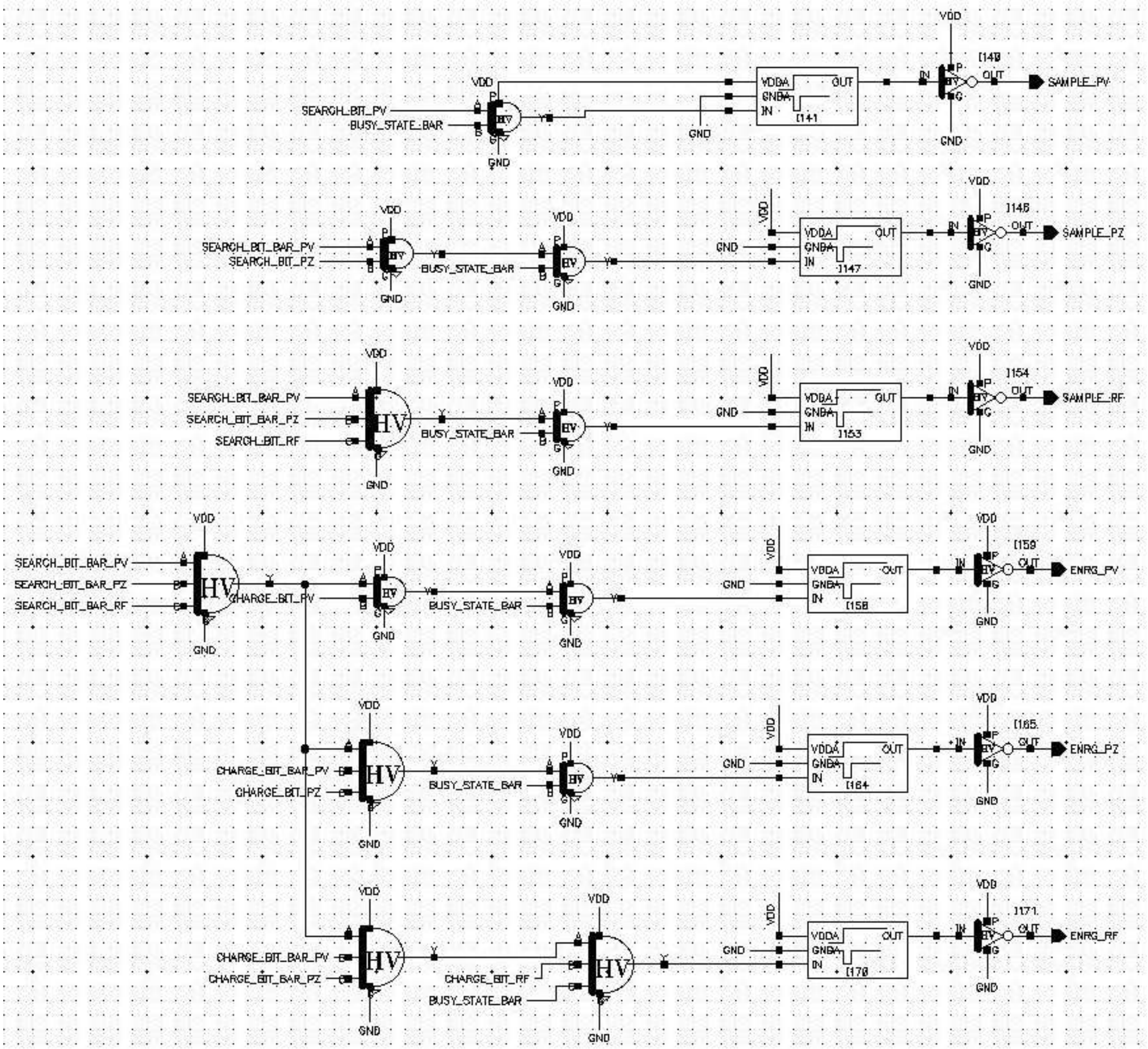

Figure 45: Generation of SAMPLE and ENRG signals

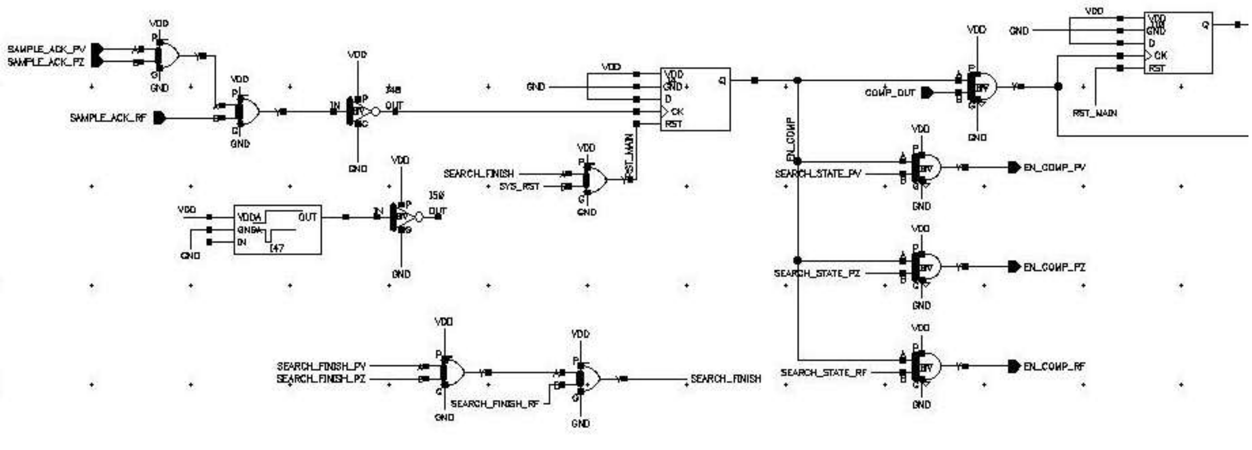

Figure 46: Generation of EN_COMP signals & starting the search procedure

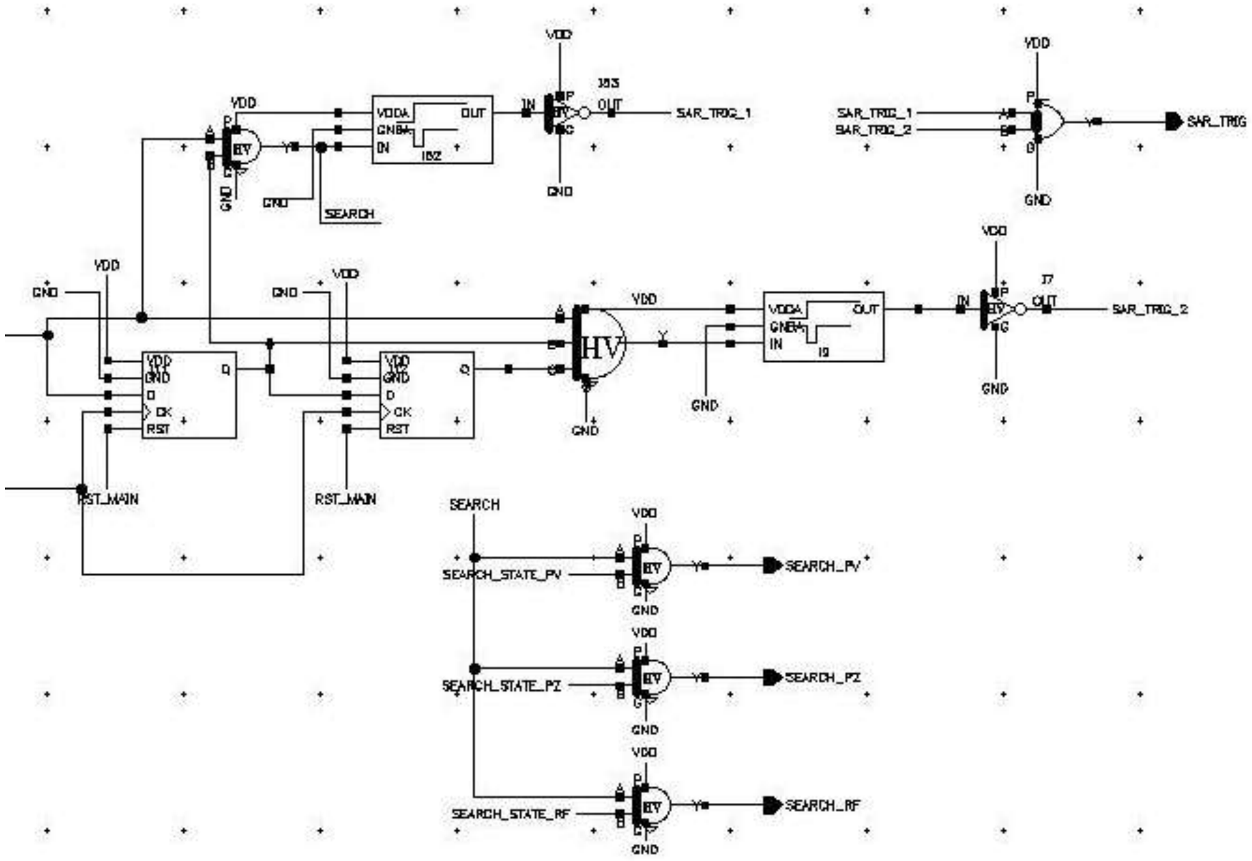

Figure 47: Generation of SAR_TRIG signal to trigger the search operation

This concludes our design for a controller with multiple sources, implementing SAR search algorithm for maximum power-point tracking.

7. Conclusion

The existing work on DC-DC converter & DTPA was studied & analyzed. To summarize, the DTPA offers no additional benefit if integrated with the existing architecture of the DC-DC converter. Rather, it only eats up the existing charge on the output cap in the boosting process & provides no extra charge to the output cap. Hence, the idea of integrating the DTPA with the DC-DC converter would not work until a method can be found to lower the charge needed during boosting or some topology can be created, which allows an alternate mechanism for boosting.

Next, an existing technique for Maximum Power-Point Tracking was thoroughly analyzed and a search algorithm was developed to replicate the functionality with a digital controller. Since the comparator consumes a huge amount of power, a digital design realized by low-powered devices was proposed as a better alternative to the same for the purpose of energy conservation. Further, the digital controller design was extended to three independent sources while sharing the maximum hardware possible in order to extract more power simultaneously from the ambient environment while not wasting a lot on extra hardware. This also reflects upon the modularity of the digital design & demonstrates how it can be extended for multiple sources painlessly.

8. Bibliography

1. Energy Harvesting. *Wikipedia, The free Encyclopedia.* [Online] September 5, 2013. http://en.wikipedia.org/wiki/Energy harvesting.

2. *Discrete-Time Parametric Amplification Based on a Three-Terminal MOS Varactor: Analysis & Experimental Results.* **Ranganathan, Sanjeev and Tsividis, Y.** IEEE Journal of Solid-State Circuits, vol. 38, Issue: 12.

3. **Tsividis, Y.** *Operation and Modeling of the MOS Transistor.* 2nd. New York : McGraw-Hill, 1999.

4. **Chowdary, Gajendranath and Chatterjee, Shouri.** *A 300-nW sensitive 50-nA DC-DC converter for energy harvesting applications.* 2014.