

ANALYZE AND DESIGN OF HIGH SPEED ENERGY EFFICIENT PULSED LATCHES BASED SHIFT REGISTER FOR ALL DIGITAL APPLICATION

Nandhini.G.S¹, PG Student, Dept. of ECE, Shree Venkateshwara Hi-Tech Engineering College, Gobi, India.

Gunasekar. N.², Asst. Professor Dept. of ECE, Shree Venkateshwara Hi-Tech Engineering College, Gobi, India.

Abstract---The area and power consumption are reduced by replacing flip-flops with pulsed latches. This method solves the timing problem flanked by pulsed latches through the use of multiple non-overlap delayed pulsed clock signals instead of the conventional single pulsed travel clock signal. The shift register uses a small number of the pulsed clock signals by grouping the latches to more than a few sub shifter registers and using additional temporary storage space latches. The implementation of FIR filters on FPGA based on established method costs all-embracing hardware recourses, which goes aligned with the decrease of circuit balance and the increase of system speed. A new design and implementation of FIR filter using Distributed Arithmetic is provide in this paper to decipher this predicament. Distributed Arithmetic structure is used to amplify the resource usage while pipeline structure is also used to increase the arrangement speed. In addition, the divided LUT method is also used to decrease the required reminiscence units. The simulation results indicate that FIR filters using Distributed Arithmetic can work stable with high speed and can set aside generally 50 percent hardware recourses to decrease the circuit balance and can be applied to a multiplicity of areas for its grand flexibility and high steadfastness.

Keywords--area-efficient, flip-flop, pulsed clock, pulsed latch, Shift register.

1. INTRODUCTION

A SHIFT register is the basic building block in a VLSI circuit. Shift registers are frequently used in many applications, such as digital filters [1], communication receivers [2], and image processing ICs [3]–[5]. Recently, as the size of the image data continue to increase due to the high demand for high quality image data, the word length of the shifter register increases to development large image data in figure processing ICs. An reflection-extraction and vector generation VLSI chip uses a 4K-bit shift register [3]. A 10-bit 208 channel output LCD column driver IC uses a 2K-bit shift register [4]. A 16-megapixel CMOS image sensor uses a 45K-bit shift register [5]. As the word duration of the shifter catalogue increases, the area and power consumption of the shift register become important design

considerations. The architecture of a shift register is quite simple. An N-bit shift register is composed of series connected N data flip-flops. The speed of the flip-flop is less important than the area and power consumption because there is no circuit between flip-flops in the shift register. The smallest flip-flop is suitable for the shift register to reduce the area and supremacy consumption. Recently, pulse latches have replaced flip-flops in many applications, because a pulsed latch is much less important than a flip-flop [6]–[9]. But the pulsed latch cannot be used in a shift register due to the timing dilemma between pulsed latches. proposes a low-power and area-efficient shift register using pulsed latches. The shift index solves the timing problem using multiple non-overlap delayed pulsed clock signals instead of the conventional single pulsed clock signal. The shift register uses a small number of the pulsed clock signals by grouping the latches to several sub shifter registers and using additional temporary storage latches. there has been a growing trend to implement digital signal processing functions in Field Programmable Gate Array (FPGA). In this sense, we need to put great effort in designing efficient architectures for digital signal processing functions such as FIR filters, which are widely used in video and audio signal processing, telecommunications and etc. Traditionally, direct implementation of a K-tap FIR filter requires K multiply-and-collect (MAC) blocks, which are expensive to implement in FPGA due to logic complexity and resource usage. To resolve this issue, DA, which is a multiplier-less architecture. Implementing multipliers using the logic fabric of the FPGA is costly due to logic complexity and area usage, especially when the filter size is large. Modern FPGAs have dedicated DSP blocks that alleviate this problem, how ever for very large filter sizes the challenge of reducing area and complexity still remains. An alternative to computing the multiplication is to decompose the MAC operations into a series of lookup table (LUT) accesses and summations. This approach is termed distributed arithmetic (DA), a bit serial method of computing the inner product of two vectors with a fixed number of cycles. The original DA architecture stores all the possible binary combinations of the coefficients $w[k]$ of equation (1) in a memory or search for table. It is evident

that for great values of L, the size of the memory containing the pre computed stipulations grows exponentially too large to be practical. The memory size can be reduced by dividing the single large memory (2L words) into multiple smaller sized memories each of size 2k where $L = m \times k$. The memory amount can be further reduced to $2L-1$ and $2L-2$ by applying offset binary coding and exploiting resultant symmetries found in the inside of the memories. This practice is based on using 2's complement binary representation of data, and the data can be precomputed and stored in LUT. As DA is a very efficient solution especially suited for LUT-based FPGA architectures, many researchers put great effort in using DA to implement FIR filters in FPGA.

2. PREVIOUSLY PROPOSED ARCHITECTURE

2.1. Master-Slave Flip-Flop

A master-slave flip-flop using two latches in Figure 2.1 can be replaced by a pulsed latch consisting of a latch. As a result, the area and power consumption of the pulsed latch become almost half of those of the master-slave flip-flop. The pulsed latch is an attractive resolution for small area and low power consumption.

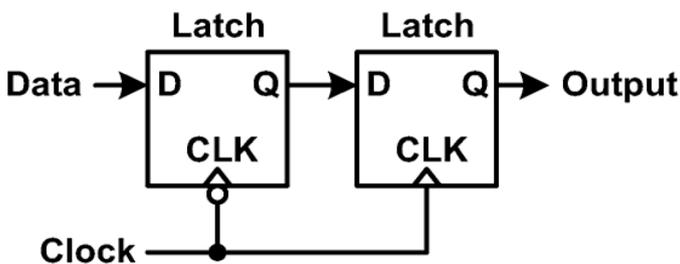


FIG.1 Master-Slave Flip-Flop

The output signal of the first latch (Q1) changes correctly because the input signals of the first latch (IN) is constant during the clock pulse width. But the second latch has an uncertain output signal (Q2) because its input signal (Q1) changes during the clock pulse width.

2.2. PULSED LATCHES

A pulsed clock signals in Figure 2.2 All pulsed latches share the pulse generation circuit for the pulsed clock signal. The pulsed latch cannot be used in shift registers due to the timing problem, as shown in Figure 3. Consists of several latches and a pulsed clock signal (CLK_pulse). One solution for the timing problem is to add delay circuits between latches.

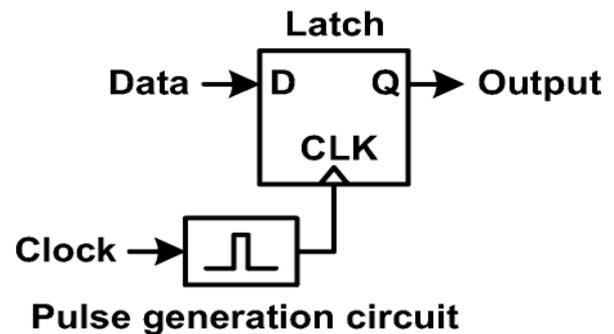


FIG.2. Pulsed latch

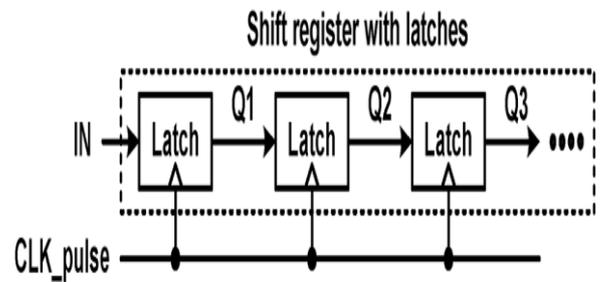


FIG.3 Shift Registers With Latches And A Pulsed Clock Signal

The shift register in Figure 3 The output signal of the latch is delayed and reaches the next latch after the clock pulse. the output signals of the first and second latches (Q1 and Q2) change during the clock pulse width, but the input signals of the second and third latches (D2 and D3) become the same as the output signals of the first and second latches (Q1 and Q2) after the clock pulse. As a result, all latches have constant input signals during the clock pulse and no timing problem occurs between the latches.

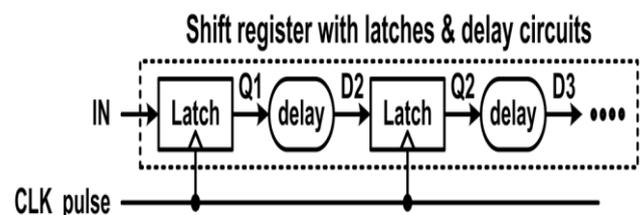


FIG.4 Shift Registers With Latches, Delay Circuits, and a Pulsed Clock

However, the delay circuits cause large area and power overheads. Another solution is to use multiple non-overlaps delayed pulsed clock signals, as shown in Figure.5.

The delayed pulsed clock signals are generated when a pulsed clock signal goes through delay circuits.

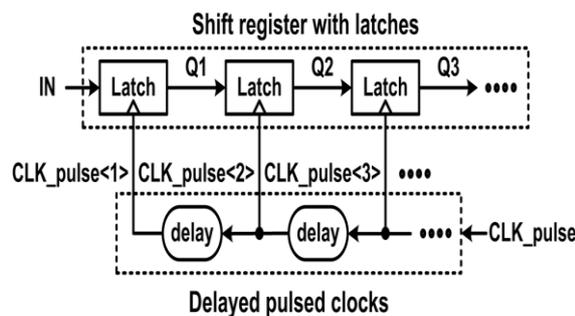


FIG.5 Shift Register with Latches and Delayed Pulsed Clock Signals

Each latch uses a pulsed clock signal which is delayed from the pulsed clock signal used in its next latch. Therefore, each latch updates the data after its next latch updates the data. As a result, each handle has a constant input during its clock pulse and no timing problem occurs between latch. However, this solution also requires many delay circuits.

The shift register is separated into sub shifter registers to reduce the number of delayed pulsed clock signals. A 4-bit sub shifter register consists of five latches and it perform shift operations with five non-overlap delayed pulsed clock signals. In the 4-bit sub shift register, four latches store 4-bit data (Q1-Q4) and the last latch stores 1-bit temporary data (T1) which will be stored in the first latch (Q5) of the 4-bit sub shift register.

Five non-overlaps delayed pulsed clock signals are generated by the delayed pulsed clock generator in Figure.3.6. The sequence of the pulsed clock signals is in the opposite order of the five latches.

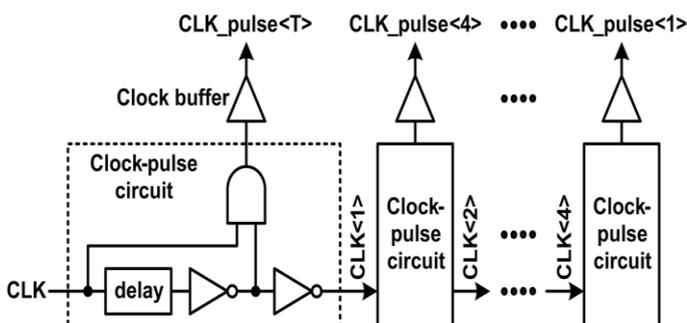


FIG.6. Delayed Pulsed Clock Generator

Initially, the pulsed clock signal CLK_pulse updates the latch data T1 from Q4. And then, the pulsed clock signals CLK_pulse update the four latch data from Q4 previous latches Q1-Q3 but the first latch Q1 receives data from the input of the shift register (IN).

The operations of the other sub shift registers are the same as that of the sub shift register except that the first latch receives data from the temporary storage latch in the previous sub shift register. To Q1 sequentially. The latches Q2-Q4 receive data from their but it increases the number of latches because of the additional temporary storage latches. As shown in Figure.2.6 each pulsed clock signal is generated in a clock-pulse circuit consisting a delay circuit and an AND gate.

In the conventional delayed pulsed clock circuits, the clock pulse width must be larger than the summation of the rising and falling times in all inverters in the delay circuits to keep the shape of the pulsed clock. The clock pulsed width can be shorter than the summation of the rising and falling times because each sharp pulsed clock signal is generated from an AND gate and two delayed signals. Therefore, the delayed pulsed clock generator is suitable for short pulsed clock signals.

The numbers of latches and clock-pulse circuits' change according to the word length of the sub shift register. Is selected by considering the area, power consumption, speed. The area optimization can be performed as follows. When the circuit areas are normalized with a latch, the areas of a latch and a clock-pulse circuit are 1 respectively.

3. PROPOSED SHIFT REGISTER

In digital circuits, a shift register is a cascade of flip flops, sharing the same clock, which has the output of anyone but the last flip-flop connected to the "data" input of the next one in the chain, resulting in a circuit that shifts by one position the one-dimensional "bit array" stored in it, shifting in the data present at its input and Shifting out the last bit in the array. Shift registers are a type of sequential logic circuit, mainly for storage of digital data.

They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. Most of the registers possess no characteristic internal sequence of states. All flip-flops are driven by a common clock, and all are set or reset simultaneously.

3.1 Storage Capacity

The storage capacity of a register is the total number of bits (1 or 0) of digital data it can retain. Each stage (flip flop) in a shift register represents one bit of storage capacity. Therefore the number of stages in a register determines its storage capacity. The serial in/serial out shift register accepts data serially – that is, one bit at a time on a single line. It produces the stored information on its output also in serial form.

A basic four-bit shift register can be construct using four D flip-flops. The operation of the circuit is as follows.

- The register is first cleared, forcing all four outputs to zero.
- The input data is then applied sequentially to the D input of the
- First flip-flop on the left (FF0).
- for the duration of each clock pulse, one bit is transmitted from

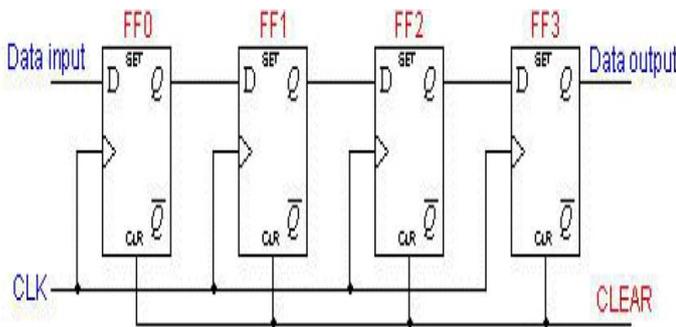


FIG.7. Four-Bit Shift Register

Above we show a block diagram of a serial-in/serial-out shift register as shown in figure.4.2, which are 4-stages long. Data at the input will be delayed by four clock periods from the input to the output of the shift register. Data at "data in", above, will be present at the Stage A output after the First clock pulse.

After the second pulse stage A data is transferred to stage B output, and "data in" is transferred to stage A output. After the third clock, stage C is replaced by stage B; stage B is replaced by stage A; and stage A is replaced by "data in". After the fourth clock, the data originally present at "data in" is at stage D, "output". The "First in" data is "First out" as it is shifted from "data in" to "data out".

For a K-Tap FIR filter the present input and past K-1 inputs must be available. For the FIR filter the input has been given serially bit wise. In the shift register block there are N-shift register which holds the K-inputs needs by FIR filter. For every clock one bit of Next input will be reaching shift register module.

So to accommodate the new bit $x[n]$ shift register must be shifted right. So for every 'B' Clock cycles where as 'b' is the no. of bits in each input sample, now input will be stored $x[n]$ and old $x[n]$ will be moved to $x[n-1]$, $x[n-1]$ to $x[n-2]$ --- like that.

That it is promising to store binary data within solid-state devices. Those storage "cells" within solid-state memory devices are easily addressed by driving the "address" lines of the device with the proper binary value(s). Suppose we had a ROM memory circuit written, or programmed, with confident data, such that the address lines of the ROM served as inputs and the data lines of the ROM served as outputs, generating the characteristic response of a particular logic function. Theoretically, we could program this ROM chip to emulate whatever logic function without having to alter any wire connections or gates.

Consider the following example of a 4 x 2 bit ROM memory (a very small memory!) programmed with the functionality of a half adder. If this ROM has been written with the above data (representing a half-adder's truth table), driving the A and B address inputs will cause the respective memory cells in the ROM chip to be enabled, thus outputting the analogous data as the Σ (Sum) and Cout bits.

Unlike the half-adder circuit built of gates or relays, this device can be set up to perform any logic function at all with two inputs and two outputs, not just the half-adder function. To change the logic function, all we would need to do is write a different table of data to another ROM chip. We could even use an EPROM chip which could be re-written at will, giving the ultimate flexibility in function.

It is vitally important to recognize the significance of this principle as applied to digital circuitry. Whereas the half-adder built from gates or relays processes the input bits to arrive at a specific output, the ROM simply remembers what the outputs be supposed to be for any given combination of inputs.

This is not much different from the "times tables" memorized in grade school: rather than having to calculate the product of 5 times 6 ($5 + 5 + 5 + 5 + 5 + 5 = 30$), school-children are taught to remember that $5 \times 6 = 30$, and then expected to recall this product from memory as needed. Likewise, rather than the logic function depending on the

functional arrangement of hard-wired gates or relays (hardware), it depends solely on the data written into the memory .The memory device simply "looks up" what the output(s) should to be for any agreed combination of inputs states.

This application of a memory device to perform logical functions is significant for several reasons:

- Software is much easier to change than hardware.
- Software can be archived on various kinds of memory media (disk, tape), thus providing an easy way to document and manipulate the function in a "virtual" form; hardware can only be "archived" abstractly in the form of some kind of graphical drawing.
- Software can be copied from one memory device (such as the EPROM chip) to another, allowing the ability for one device to "learn" its function from another device.
- Software such as the logic function example can be premeditated to perform functions that would be extremely difficult to emulate with discreet logic gates (or relays!).

The usefulness of a look-up table becomes more and more evident with increasing involvedness of function. Suppose we wanted to build a 4-bit adder circuit using a ROM. We'd require a ROM with 8 address lines (two 4-bit numbers to be added together), plus 4 data lines (for the signed output): In our design we are using look table to store all the different possible combination summation of filter coefficients.

In the reset state it will be store the coefficient summations. Input for this LUT will be coming from the output of shift register module. For every clock cycle the LSB bits of all 'N' input samples are applied to the LUT.LUT will consider this input as on address and bestow the data stored in that particular address to the output. The main improvement of LUT realization is we can stay away from the multiplications.

4. RESULT AND DISSCUTIONS

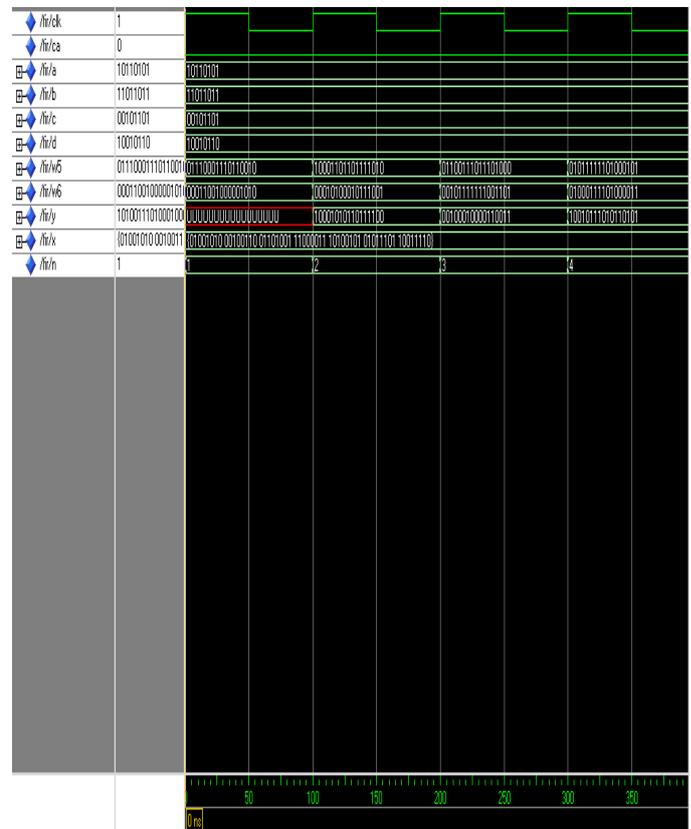


FIG.8.Diagram of output

5. CONCLUSION

The shift register reduces area and power consumption by replacing flip-flops with pulsed latches. The timing difficulty between pulsed latches is solved using multiple non-overlap deferred pulsed clock signals instead of a single pulsed clock signal. A small number of the pulsed clock signals are used by grouping the latches to several sub shifter registers and using additional temporary storage latches. The design and implementation based on Distributed Arithmetic, which is used to realize a 31-order FIR low-pass filter. Distributed Arithmetic structure is used to amplify the recourse tradition while pipeline structure is used to increase the system speed. The test results indicate that the designed filter using Distributed Arithmetic can work stable with high speed and can put away almost 50 percent hardware recourses. Meanwhile, it is very easy to transplant the filter to other applications from beginning to end modifying the order parameter or bit width and otherparameters, and therefore have large practical applications in digit signal processing.

ACKNOWLEDGEMENT

We are expressing our thanks to all Faculty members and Skilled Assistants of Electronics and Communication Engineering department and my Friends who helped me in every possible way. Last but not least I thank my Parents for their moral support.

REFERENCES

1. Consoli,E, M. Alioto, Palumbo,G, and Rabaey,J, "Conditional push-pull pulsed latch with 726 fJops energy delay product in 65 nm CMOS,"(2012) in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, pp. 482-483.
2. Heo,S, Krashinsky,R, and Asanovic,K, "Activity-sensitive flip-flop and latch selection for reduced energy," (2007) IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 9, pp. 1060-1064,.
3. Kong. B.-S, Kim. S.-S, and Jun. Y.-H., "Conditional-capture flip-flop for statistical power reduction," (2001) IEEE J. Solid-State Circuits, vol. 36, pp. 1263-1271,
4. Kim. H.-S, Yang,J.-H, Park.S.-H, Ryu. S.-T, and Cho. G.-H, "A 10-bit column-driver IC with parasitic-insensitive iterative charge-sharing based capacitor-string interpolation for mobile active-matrix LCDs," (2014) IEEE J. Solid-State Circuits, vol. 49, no. 3, pp. 766-782,
5. Nomura,S, et al., "A 9.7 mW AAC-decoding, 620 mW H.264 720p 60fps decoding, 8-core media processor with embedded forward body-biasing and power-gating circuit in 65 nm CMOS technology,"(2008) in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, pp. 262-264.
6. Naffziger.S and Hammond.G, "The implementation of the nextgeneration 64 b titanium microprocessor,"(2002) in IEEE Int. Solid-State CircuitsConf. (ISSCC) Dig. Tech. Papers, pp. 276-504.
7. Reyes.P, Reviriego.P, Maestro. J. A, and Ruano.O, "New protection techniques against SEUs for moving average filters in a radiation environment,"(2007)IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 957-964,
8. The.C.K, Fujita.T, Hara.H, and Hamada.M, "A 77% energy-saving 22-transistor single-phase-clocking D-flip-flop with adaptive-coupling configuration in 40 nm CMOS,"(2011) in IEEE Int. Solid-State

Circuits Conf.(ISSCC) Dig. Tech. Papers, pp. 338-339.

9. Tearney G J and Bouma B E. Real-Time FPGA Processing for High-Speed Optical Frequency Domain Imaging [J]. IEEE Transactions on Medical Imaging, (2009), 28(9):1468~1472.
10. Tsao Y C and Choi K. Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, (2010), PP (99):1~5.
11. Ueda.Y et al., "6.33 mW MPEG audio decoding on a multimedia processor," (2006) in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, pp. 1636-1637.

BIOGRAPHIES



Gunasekar. N received his B.E. degree in Electronics and communication engineering from Nandha Engineering College, Erode, Tamilnadu in 2011, The M.E. degree in Applied Electronics from RVS college of Engineering and Technology, coimbatore, Tamilnadu in 2013. At present, He is engaged in Ph.D from Kongu engineeringcollege, He was an Assistant professor, Shree Venkateshwara Hi-Tech Engineering College, 2013-2016.



Nandhini.G.S received the B.E degree in electronic and communication engineering with first class from shree venketeshwara Hi-tech Engineering College, Gobi, Tamilnadu in 2014, At present, She is engaged in M.E in applied electronics from Shree Venkateshwara Hi-Tech Engineering College, 2014-2016.