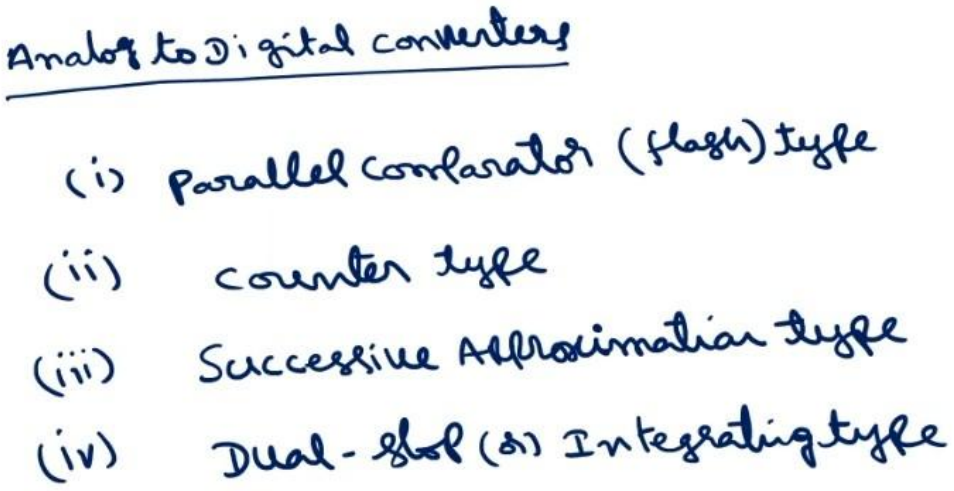
Integrated Circuits and Applications
Prof. Shaik Rafi Ahamed
Department of Electronics and Electrical Engineering Indian Institute of Technology, Guwahati

Data Converters
Lecture – 40
Analog to Digital Converters

So, in the last lectures, we have discussed various types of D to A converters. So, the other type of a data converter is a analog digital converter. If you want to process the analog signal using digital signal processor, first we have to convert this analog signal into digital signal. Because digital signal processors are having lot of advantages when compare with the analog signal processors. There are four types of analog to digital converters. The first one is called the flash type or parallel comparator type.

And second one is called counter type. Third one is successive approximation type. And fourth one is dual slop or integrating type. Each one is having its own advantages and disadvantages.



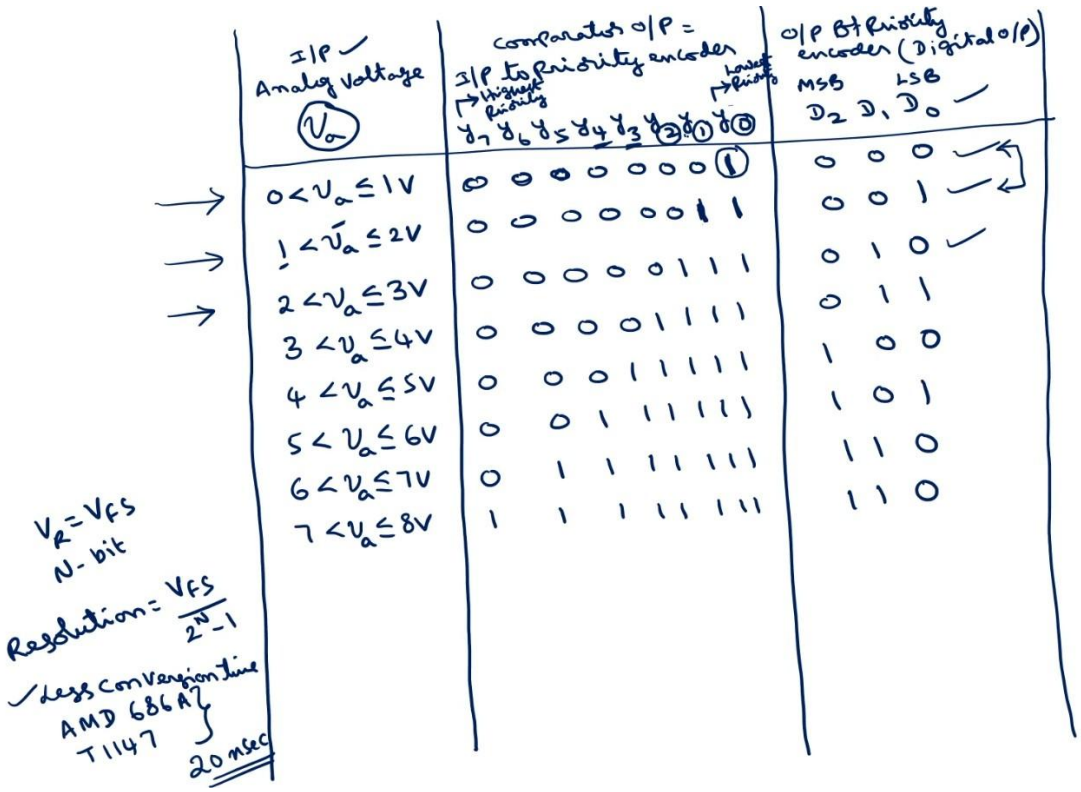
(Refer to the slide at 02:28)

So, I will discuss the circuit diagram of each of these analog to digital converters and their relative merits and demerits. The first analog to digital converter is parallel comparator type A to D converter. This is the fastest A to D converter among all the A to D converters. But the drawback of this one is more complexity. So, at the end of this discussion we can see that the complexity of this A to D

converter is more.

So, in general, here, as the name implies, the parallel comparator will be having comparators for n bit ADC contains $2^n - 1$ comparators. Here, I will discuss about the 3-bit parallel comparator type ADC. So, this consists of 7 comparators. This is the only one comparator. The positive terminal is connected to the analog signal, which is to be converted into digital and negative terminal is connected to the V reference.

Similarly, for all the comparators, positive terminal is connected to the same v_A. Negative terminal will be connected to a part of reference voltage, which we can obtain by using resistor. Here all these v_A 's will be connected together. This is the analog voltage which is to be converted into digital. Here, V_R and bit fin the two consecutive inverting terminals, there will be resistors.



(Refer to the slide at 18:54)

This is R, R, R. So, this voltage will be V_R directly. What will be voltage here? V_R into the total resistors below this point is 1, 2, 3, 4, 5, 6 and total resistors are 7. So, kind of voltage divider $V_R \times \frac{6R}{7R} = \frac{6}{7}v_R$. Similarly, here, this voltage will be $\frac{5}{7}V_R$.

This will be $\frac{4}{7}V_R$. This will be $\frac{3}{7}V_R$. This will be $\frac{2}{7}V_R$. This will be $\frac{1}{7}V_R$. For the sake of simplicity, if I take $v_R = 7V$, and this will be 7V, this will be 6V, this will

be 5V, this will be 4V, 3V, 2V, 1V.

Then, this output of these comparators will be applied to a priority encoder. So, we know that the encoder will be having 2^n inputs and n outputs. If I call this output as Y7 MSB, Y6, Y5, Y4, Y3, Y2, Y1, but if you want to use a 8 to 3 priority encoder, we require one more input. So, I will connect this Y0 permanently to logic 1 and displace 5V logic 1. This will produce 3 outputs a D_2, D_1, D_0, D_2 is MSB, and D_0 is LSB.

Now, the operation of this circuit is, if this analog voltage v_A is less than or equal to 1V, this 1V. So, it will be output of the comparator. This voltage is less than or equal to 1V; here, the voltage is 1V. So, the voltage at negative terminal is greater than the voltage at positive terminal, or if both the inputs are equal, then also output is equal to 0. Then the output Y1 becomes 0, and what happens to Y2? This voltage is less than or equal to 1V, whereas this voltage is 2V. So, Y2 also will be 0, Y3 is 1 up to all Y7 all will be 0 because the comparator operation is something like this.

This is A, this is B. If $A \geq B$, implies output if I call as Y, Y is equal to because negative terminal voltage is greater than B. So, output is equal to 0; if $A < B$ or $B > A$, output is equal to logic 1. So, according to that, say, if I take between 0 to 1V, this Y1 to Y7, all will be 0; if I apply v_A, which is slightly more than 1V, then what happens here the voltage will be slightly more than 1V then the output becomes 1. Similarly, if I apply the voltage here, which is slightly more than 2V, this output is 1; if the voltage here is slightly more than 3V, this output is 1; if it is slightly more than 4V, this output is equal to 1 and so on. So, if you tabulate these value this will be something like this analog voltage which is input comparator output, which is also input to encoder now, input of priority encoder, this is also will be the digital output this is analog input, and this is corresponding digital output.

This is v_A, the analog voltage which is to be converted into digital this is D_2, D_1, D_0 in corresponding digital outputs with D_2 as MSB, D_0 as LSB in between, we have Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0, Y0 is permanently connected to 1. If $0 < v_A \leq 1V$. As you have seen here, if this voltage is less than or equal to 1V, this is also 0; this is also 0 All outputs Y1 to Y7 is 0 except Y0, this is 0 0 0 0 0 0 0, and this is permanently 1. Now, if you take the next range $1 < v_A \leq 2V$. Now, v_A is greater than 1V means the last comparator output will be 1 because this voltage is more than this voltage.

So, output is logic 1 because positive terminal voltage is less than negative terminal voltage. So, this will be 0 0 0 0 0 0 this will be 1 whereas, Y0 is already permanently 1. Now, if I take $2 < v_A \leq 3V$ then the last 3 inputs will be 1 and so on, all will be 1. Now, what is the operation of this priority encoder? Here if only one input to the priority

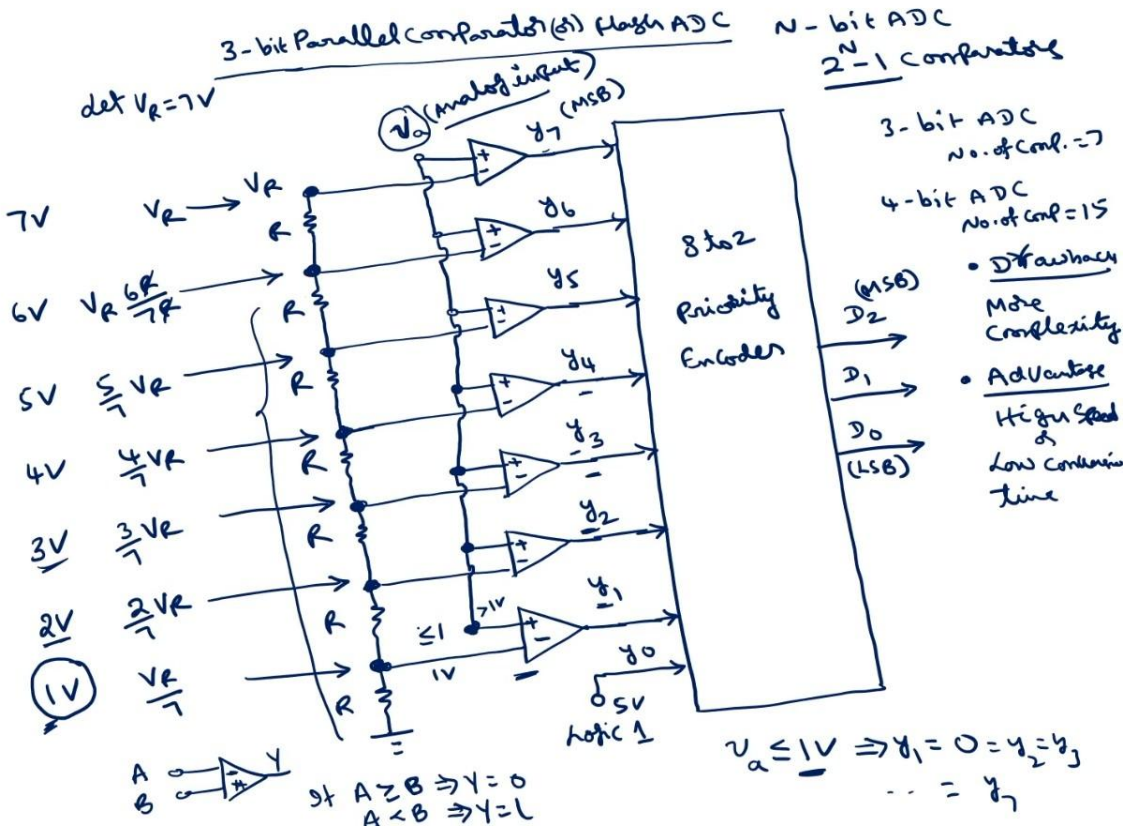
encoder is 1, then that particular input will be encoded. Now, before encoding, we have to assign the priorities. I am assuming that this Y7 is having highest priority and the next is Y6, Y5, and so on Y0 is having lowest priority.

Now, for the first combination of the inputs, only one input is 1, which is the lowest priority input. So, that Y0 will be encoded in the sense we are going to write the 3-bit digital equivalent of this suffix 0 3 bit equivalent is 0 0 0. Now, in the second combination, 2 inputs are 1. So, between these 2 the input with the larger priority will be encoded between Y1 and Y0 Y1 is having larger priority. So, this 1 if I take as decimal value, the binary equivalent 3 bit binary equivalent of 1 is 0 0 1.

So, in this case this is both are 1s 3 inputs are 1. So, Y2 will be encoded because among these 3 Y2 is having largest priority. So, 2 binary equivalent will be 0 1 0. Now, here Y3 3 means 0 1 1 here Y4 4 means 1 0 0 next 5 next 6 next 7. This is how we can convert this voltage range into 0 0 0, this voltage range into 0 0 1 this voltage range into 0 1 0 so on.

So, the given analog voltage is converted into corresponding digital output. The minimum input voltage that is required to change from one code word to another code word is slightly more than 1V ok. So, in fact, this is also similar to the resolution of D to A converter here also if V_R or V full scale is the full scale voltage of A to D converter for N bit A to D converter the resolution. The minimum change in the input required to change one code word is equal to $\frac{V_{FS}}{2^N-1}$. So, this is the operation of parallel comparator or flash type ADC.

The advantage of this type of ADC is you see having less conversion time. The time taken to convert the analog signal into digital signal is very less. So, here, if I use the high speed comparators and a priority encoders, the comparator of the form of AMD companies 686A and if I use the priority encoder T1147, then the conversion time of this type of ADC is of the order of 20 nanoseconds, which is very less. But the drawback of this type of ADC is the complexity is more for every one increase in the bit the number of comparators will be almost doubles. We can see that for a 3-bit ADC, the number of comparators are 7; for 4 bit ADC, the number of comparators will be $2^N - 1$, which is 15. This is almost double ok.



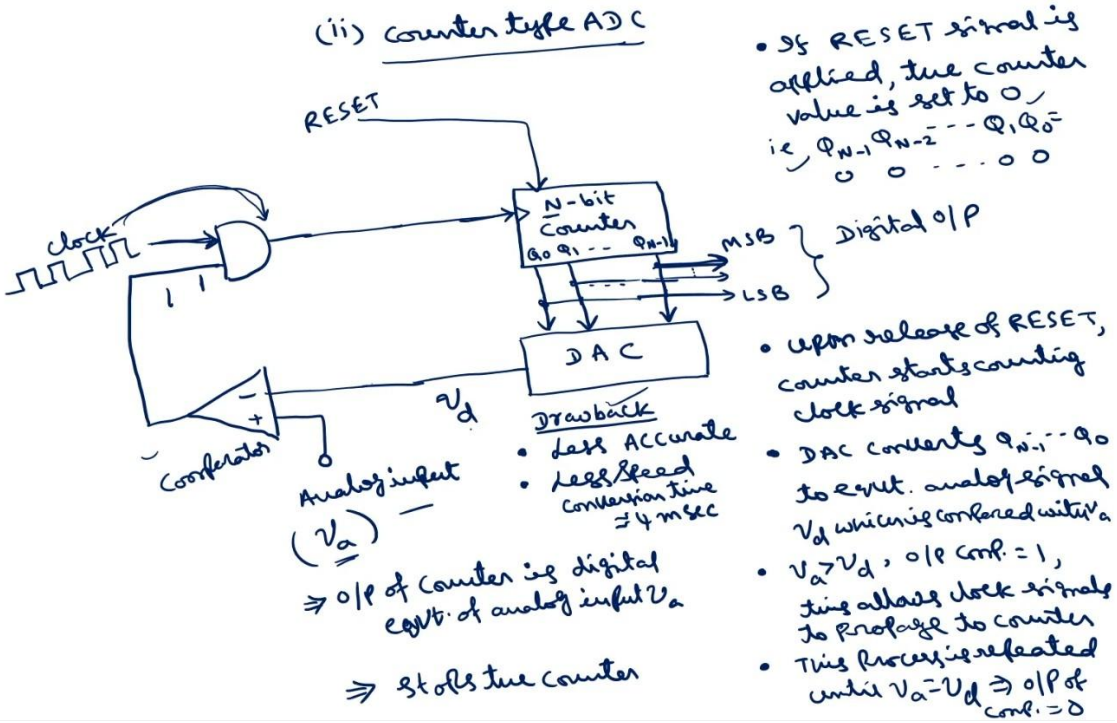
(Refer to the slide at 20:14)

So, this is the drawback: drawback is complexity more and as advantage is high speed or low conversion time. So, in applications where speed is important, we will prefer this parallel comparator or flash type ADC. So, second type of ADC is called as a counter type ADC. As the name implies there will be a counter which counts the clock signals. So, there is a counter, say N bit counter this will give n outputs Q_0, Q_1 so on up to Q_N minus 1.

So, inside this counter there will be flip flops output of the flip flop will be Q_0 to Q_N minus 1. So, this is MSB and this is LSB, MSB, LSB. So, this will be applied to D to A converter, this will give some digital output say v_d is the digital output correspond to this analog input. Correspond to v_d is let v_d is the analog output correspond to this digital input. And this will be compared with the analog signal, which is to be converted.

Here, we are going to apply analog input v_A, which is to be converted into digital. So, the corresponding digital output we are going to take here. This comparator output is applied to the AND gate, whose other input is clock signal. So, we know that for the AND gate, if both the inputs are 1, then only output is 1. So, if the output of the comparator is 1, then only this clock signal will be propagated to the output.

This is going to enable the clock of the counter and there will be some reset signal for this counter. This is the block diagram of counter type ADC. So, the operation is if reset signal is applied. The counter value is set to 0, that is, Q_{N-1}, Q_{N-2}, 1 up to Q_1, Q_0 will be all 0s. Upon the release of this reset signal, counter starts counting the clock signals.



(Refer to the slide at 28:18)

So, this Q_{N-1} to Q_0 becomes 1, 2, 3 up to 2^N. Then this D to A converter converts Q_{N-1} up to Q_0 to equivalent digit equivalent analog signal, which is v_d. Now, we have to compare this v_d with v_A using comparator which is compared with v_A. As long as $v_A > v_d$, output of comparator will be 1. So, this allows because this input is 1.

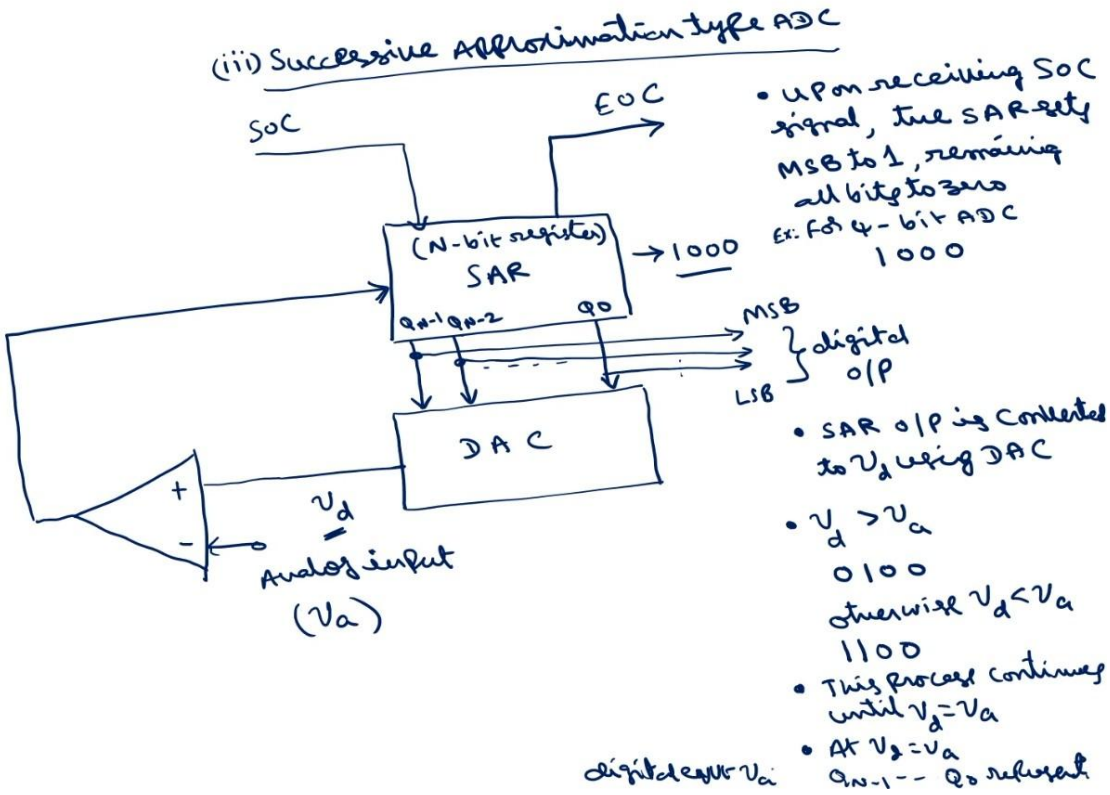
So, for AND gate output will be simply this clock signals. So, this clock signals will be applied to the counter. Now, whole counter will increment the count, so, like that. So, this process will repeats until $v_A = v_d$. Once $v_A = v_d$, implies output of the comparator is 0, then stops the counter.

So, at this time whatever the output of this counter is digital equivalent of analog signal. This is the operation of this counter type ADC, but the drawback of this one is because of

this DAC and comparator, this is less accurate. And, also takes more conversion time less speed. Conversion time will be of the order of 4 milliseconds which is much greater than that of the flash type ADC whose conversion time is approximately 20 nanoseconds. So, because of these reasons, this counter type ADC is not popular.

So, the third type of ADC is successive approximation type ADC. This is also similar to this counter type ADC; here, there will be a successive approximation register SAR is successive approximation register. This will be having two signals one is called start of conversion another is end of conversion. So, this is N bit register. So, each register will be having one output, this also will give N outputs.

Q_{N-1} is MSB Q_{N-2}, 1 up to Q_0. This will be applied to the D to A converter, similar to the counter type ADC. This will produce analog voltage AD, which is proportional to this digital. This is MSB, this is LSB, this is digital output. And, here also this will be compared with the comparator. Here we are going to apply the analog signal which is to be converted into digital.



(Refer to the slide at 33:55)

This is going to control the successive approximation register. Here, the operation is whenever start of conversion normally received by this SAR upon receiving start of conversion signal. The SAR sets MSB to 1 and remaining all bits to 0. And, if it is 4-bit ADC, this will make 1 0 0 0. So, the initial value of this A to D converter, this successive approximation register the initial value is 1 0 0 0.

Now, this value of SAR output is converted to v_d using DAC. Now, this v_d is proportional to this digital value of this SAR. Now, if v_d is here, there are two cases: v_d can be greater than VAR, v_d can be less than v_A. If $v_d > v_A$, what happens is then this first MSB bit will reset, next MSB bit will reset to 1. If otherwise means if $v_d < v_A$, then the first MSB bit will keep 1 the second MSB bit also will becomes 1 remaining it will be 0.

So, this process continues until $v_d = v_A$. Once $v_d = v_A$, this SAR, whatever the output of SAR, will be the digital equivalent of the analog input. This Q_{N-1} so on up to Q_0 represents digital equivalent of analog signal v_A this is the operation. This will be better understand if I take an example of 4-bit ADC. So, the first value of 4-bit ADC is 1 0 0 0. From this, we have two options.

If this v_d correspond to this one v_d, if this $v_d > v_A$, then first MSB bit will be reset, next MSB bit will be set. If $v_d < v_A$, then first MSB bit was 1. We will set second MSB bit also we will keep the first MSB bit 1, and we will set the second MSB bit also 1 0 0. Again, here there are two options depends upon whether this analog equivalent voltage v_d is greater than or less than v_A what are the two options this can be 0 0 1 0 or 0 1 1 0. Again, there are two options here: that we have to keep this MSB bit or not depends upon v_A is greater than or less than. So, 0 0 0 1 if $v_d > v_A$, if $v_d < v_A$ then this will be 0 0 1 1.

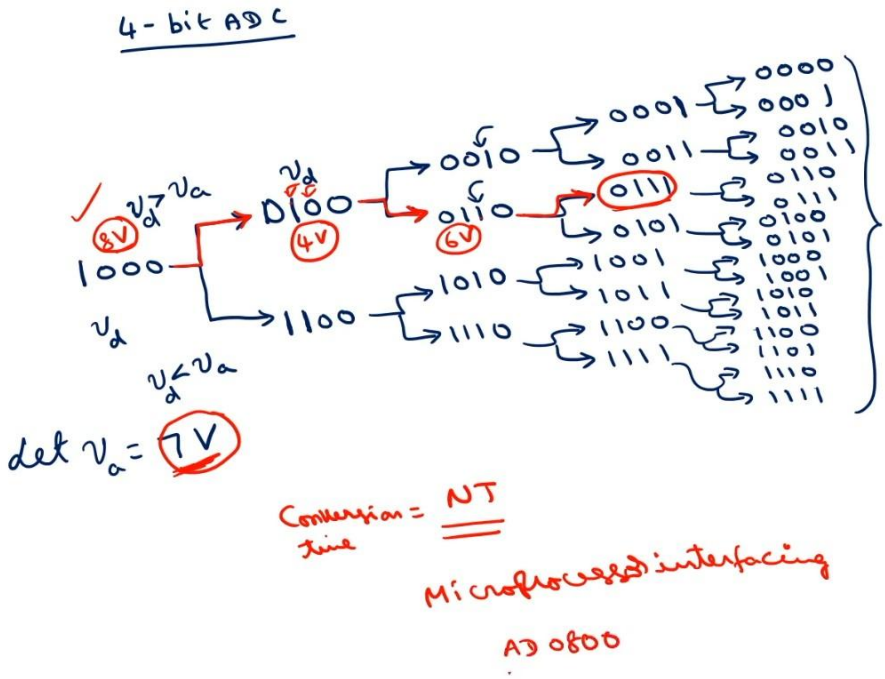
Here also we have two options what are these two options 0 1 1 1 and 0 1 0 1. Whether, you have to keep this bit or you have to reset this and you have to set the next bit depends upon v_d is greater than or less than. Similarly, here also, we will be having two options 1 0 1 0, 1 1 1 0. Similarly, here also, we have two options: 1 0 0 1, 1 0 1 1. Here also we have two options 1 1 0 0, 1 1 1 1. Here also we have now two options one is we have to keep this or we have to reset this 0 0 0 0, 0 0 0 1.

Here, also we have to keep this 0 or 1. So, this is 0 0 1 0, 0 0 1 1, 0 1 1 0 or 0 1 1 1. This is 0 1 0 0, 0 1 0 1, 1 0 0 1, 0 1 0 1, 0 1 0 1 1. This will be 1 0 1 0, 1 0 1 1 this will be 1 1 0 0 or 1 1 0 1 this will be 1 1 1 0, 1 1 1 1 totally we have 16 combinations. Let us take, let $v_A = 7V$, then what will be the path? So, initial value of this one is digital equivalent is 8V.

So, compared to 7V, this is greater. So, what we will do? So, we will reset this we will go this is the path because this is greater than the analog voltage. So, we will reset the first bit and we will set the next bit. Now, what is this analog value 4V? This is less than 7V? So, we will keep this bit as 1, and we will set the next bit, also means after that, it will go here.

Now this is 6V this is less than this. So, we will keep the third bit also and we will make the fourth bit also 1. So, then after that this will go here. So, this will be the digital equivalent of this 6V. Like, that you can take any analog voltage, you can follow the flow to get the corresponding digital equivalent of this given analog signal.

This is how this 4-bit ADC will works. So, this ADC is having a conversion time of NT; here if T is the clock period N is the number of bits in the ADC. So, this speed is intermediate to this flash type and counter type at the same time complexity also mode rate because of that this type of ADC will be mostly used in microprocessor interfacing. This ADC is available in a monolithic form in the form of a IC. We have AD0800 series. So, the last type of ADC is integrating type or dual slope type.



(Refer to the slide at 39:18)

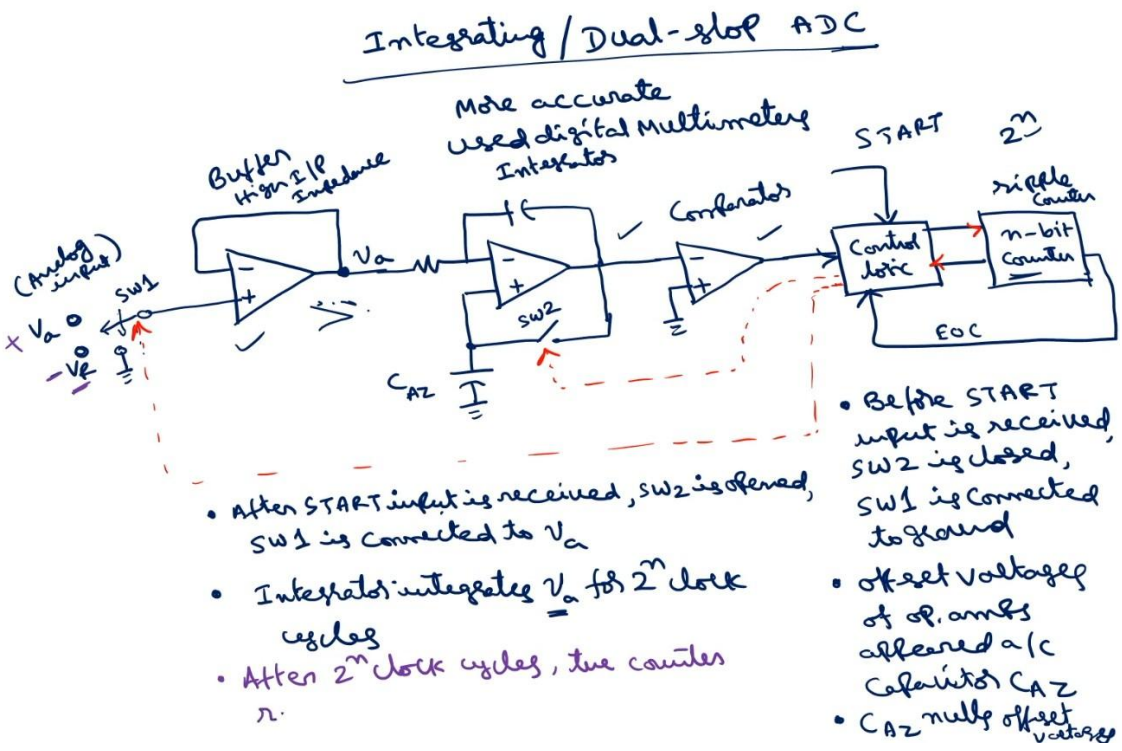
This is the more accurate ADC compared with the all other ADCs. So, this is more accurate because of that this will be used in a digital multimeters. So, this is basically consists of a buffer. Then followed by integrator. This is switch 1, this is switch 2 and applied to the comparator.

Compilator output will be applied to the control logic. This control logic is connected to the counter. End of conversion. This control logic controls the switch positions. This is a typical block diagram of this one.

This will acts as a buffer is used to provide high input impedance. This will be integrator, and this is comparator. There is control logic counter. This is basically ripple counter. So, the count will be 2^n because n bit counter. So, the operation is initially before the start pulse is applied, we have the start here.

We will close this SW2 switch. SW2 is closed and SW1 is connected to ground. Here, we have three positions: one is minus V_R, one is v_A, another is ground. This switch will be connected to this ground.

You know this switch has two SW2 is closed. So, the capacitor charges. So, all the offset voltage of these three circuits will be appeared across capacitor. So, this CAZ will be basically used to null the offset voltages. So, when the START input is received SW2 is opened. So, that CAZ will nullify the offset voltages and as SW1 is connected to v_A, the analog voltage which is to be converted. So, here, this voltage also v_A, then the integrator integrates v_A per 2^n clock cycles.



(Refer to the slide at 47:53)

Because this counter, after counting 2^n clock signal, this will reset. So, this v_A will be integrated over the 2^n clock cycles. So, if I take this waveform, this integrator output say, at this point, the start has been received. So, this is t_1, then this will integrate, here start is applied.

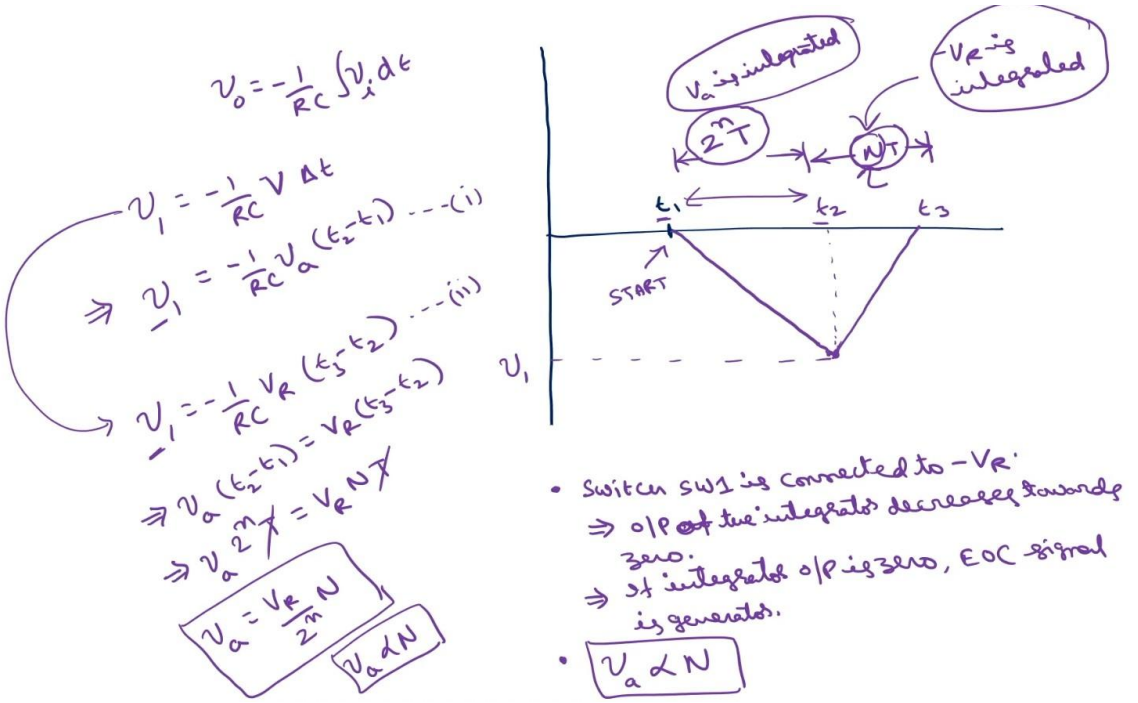
So, within this duration the integrator integrates v_A. For how many clock cycle? $2^n \times T$ is the period. If T is the clock period, $2^n \times T$ is the period between t_1 to t_2. So, after this 2^n clock signals, the ripple counter will reset, then we will apply this switch position to $-V_R$. Now, because these two are of opposite polarity, this is plus, and this is minus, then the output of this integrator will be in the reverse direction.

Now, this will decrease in this fashion. Let us assume that this will come to the 0 for n clock signals. So, after 2^n clock cycles, the counter resets, then the switch SW1 is connected to minus V_R. So, implies the output of the integrator decreases towards 0. So, if the output of the integrator is 0, you have to stop.

End of conversion, signal is generated. Now, we can show that this n over which the V_R is integrated till the output of the integrator is 0; this $v_A \propto N$. We are going to show that this $v_A \propto N$. How to show this? So, we know that the output of the integrator is equal to

$-\frac{1}{RC}\int v_i dt$. If I assume that this is v_1, then $v_1 = -\frac{1}{RC} v\Delta t$.

If I consider this delta t from t_2 to t_1, what will be this V? V will be v_A. So, $v_1 = -\frac{1}{RC} v_A(t_2 - t_1)$. Between this interval, v_A is integrated. Between this integral -V_R is integrated. So, this is one expression.



(Refer to the slide at 52:03)

Again, using the same because this v_1 is common for both the v_A as well -V_R. This $v_1 = -\frac{1}{RC} V_R(t_2 - t_1)$. This is decreasing; I am taking t_3 to t_2, I am writing the positive value. This is equation 2 because this is also v_1, this is also v_1; you can equate these two. So, if we equate these two, what happens? Minus 1 by RC minus 1 by RC will get cancelled plus you will get $v_A(t_2 - t_1) = V_R(t_3 - t_2)$.

What is t_2-t_1? $2^n T$, and t_3 - t_2 is NT. So, this implies $v_A 2^n T = V_R NT$. So, T T get cancelled. So, $v_A = \frac{V_R}{2^n} N$. So, 2^n is constant, V_R is constant.

So, $v_A \propto N$. So, whatever this count over which this V_R is integrated, that is proportional to the given analog voltage. This is the operation of this integrating type ADC. So, this is more accurate, and this will be used in digital multimeters. So, these are all about the four

types of the ADCs. Next lecture, we will discuss about the digital ICs such as CMOS circuits. Thank you.