Computer Numerical Control of Machine Tools and Processes

Professor A Roy Choudhury
Department of Mechanical Engineering
Indian Institute of Technology Kharagpur

Lecture 13

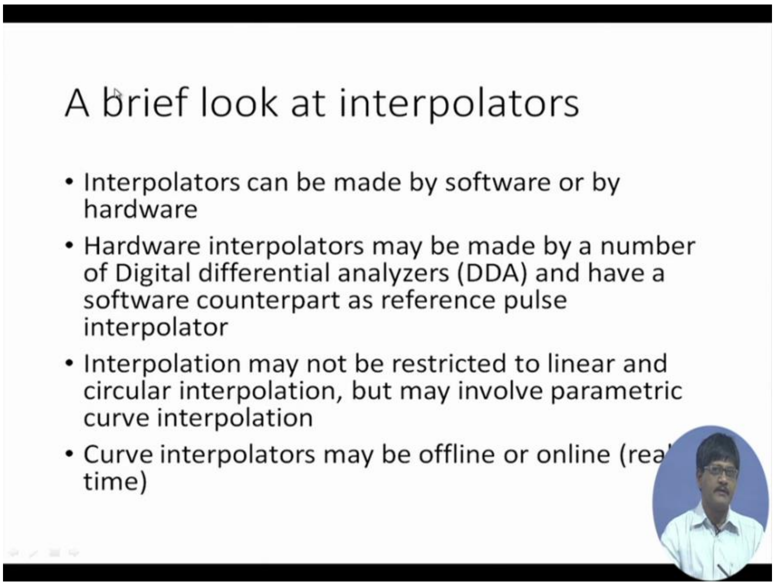
Interpolators Linear

Welcome viewers to 13th Lecture of the online course "Computer control of machine tools and processes". So in this particular lecturer we will be discussing somewhat about interpolators and to a large extent linear interpolators. What is the purpose of having interpolators? First of all, we are having a continuous control system, we are going to cut profiles and there we might have to cut straight lines, I mean we might have to cut along straight lines or along a circular arc or in a more generally we, we might have to cut along complex curves also.

Not only spatial curves, but also three-dimensional that means 3-D curves. First of all, if we have to cut along a straight line, it apparently seems to be quite an easy task okay, cutting along the straight line seems to be the easiest possible or the simplest possible movement, so why do we have to have this interpolator, linear interpolator for this? This is because in case of interpolation we are only being provided with I mean the program is only providing the machine controls with an initial point and a final point. So in between the points or the location through which the machine is I mean the cutter is going to move relative to the work piece, these are not provided.

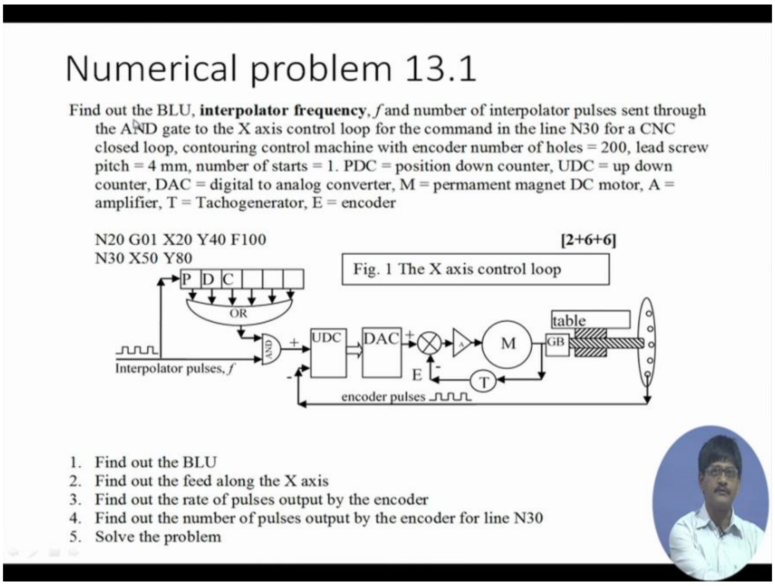
Also, the velocities that are supposed to be developed by the different axis of motion that is also not known, so this way two specific tasks of the interpolator emerges; one is the interpolator has to make the cutter maintain a definite velocity rate of velocity along the profile of the work piece, so along the path the cutter has to move at the feed rate program. Second, in order to move along the path the cutter has to have a definite ratio existing between the different axis velocities. If it is linear motion, these axis velocities are constant therefore the ratio also remains constant, this is the way in which linear interpolator can be specified, so let us have a look what we have.

(Refer Slide Time: 03:17)



So linear interpolators can be made by software or hardware and hardware interpolators may be made by a number of digital differential analyzer which we have already studied in the previous lecture on logic circuit and decoder. It is also possible to make software counterparts of this particular interpolator, which is sometimes referred to as reference pulse interpolator. So interpolation as we just now discussed, it might not always be restricted to linear and circular interpolation but also may involve parametric curve interpolation. And these curve interpolators maybe off-line or online.

(Refer Slide Time: 04:08)



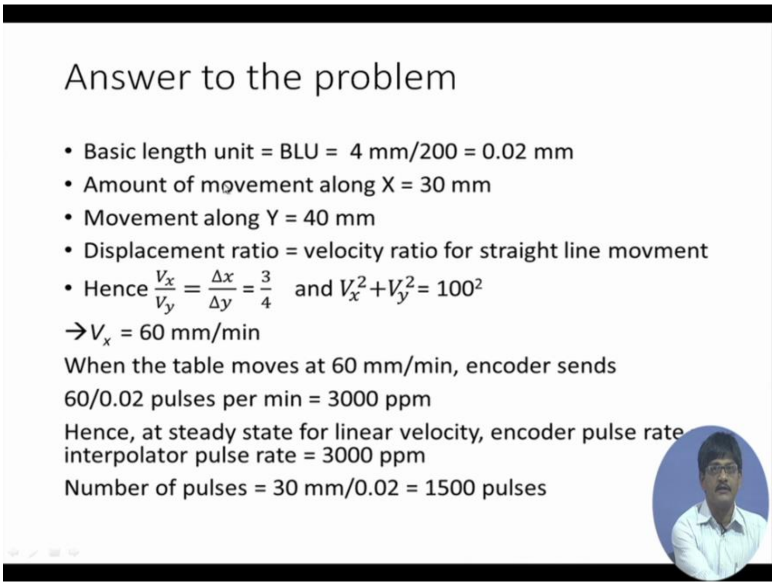
So first of all, let us take a look at a numerical problem, this will help us in establishing the different relations which exists between different parts of an interpolator, what is this problem? This problem states that find out the basic length unit, the interpolator frequency *f* and the number of interpolator pulses sent through the AND gate of the X axis control loop for the command in the line number 30 for a CNC closed loop Contouring control or continuous control machine with encoder number of holes 200, leads screw pitch 4 millimeters, number of starts = 1. There is position down counter and there is an up down counter also in the control loop, there is a digital to analog converter and M is the permanent magnet DC motor, A the amplifier, T the tachogenerator and E the encoder.

We are already conversant with the sort of control loops, I will just read it one or one or 2 very important points that is interpolator pulses are coming in and 1st of all they determine the position down counter, it is the custodian of the position. So if we only allow a specific number of pulses to be pass through, the down counter keeps track of the balance of the pulses of the interpolator and the encoder, while the interpolator is filling up the up down counter with higher and higher number, the encoder is going on the decrementing the up down counter content.

So that when the motor is rotating the lead screw at the required RPM that means the programmed RPM, at that time the down counter content becomes steady so that in linear interpolation a steady state speed is reached. This thing we have already discussed, so I do not want to elaborate on it once more, so let us concentrate upon the problem. We can formulate the problem solution this way that is we will find out the basic length unit, we will find out the feed along the X axis, we will find out the rate of pulses output by the encoder and we will find out the number of pulses output by the encoder for line number 30 and that way we will solve the problem.

What exactly is the problem? The interpolator pulse rate is required, what is the frame rate of pulses coming along the line of interpolator pulses that line which is drawn on the left-hand side, these pulses what is the frequency for this particular command? Let us have a look at the command, N20 G01 X20 Y40 F100, so the initial point for the next line would be (20, 40). From the point (20, 40) in line number 30 we are reaching X = 50 and Y = 80, which means X is moving from 20 to 50, 30 millimeters and Y is moving from 40 to 80, which means 40 millimeters, so 30 millimeters along X and 40 millimeters along Y and the feed is of 100 millimeters, so let's go to the solution.

(Refer Slide Time: 07:47)



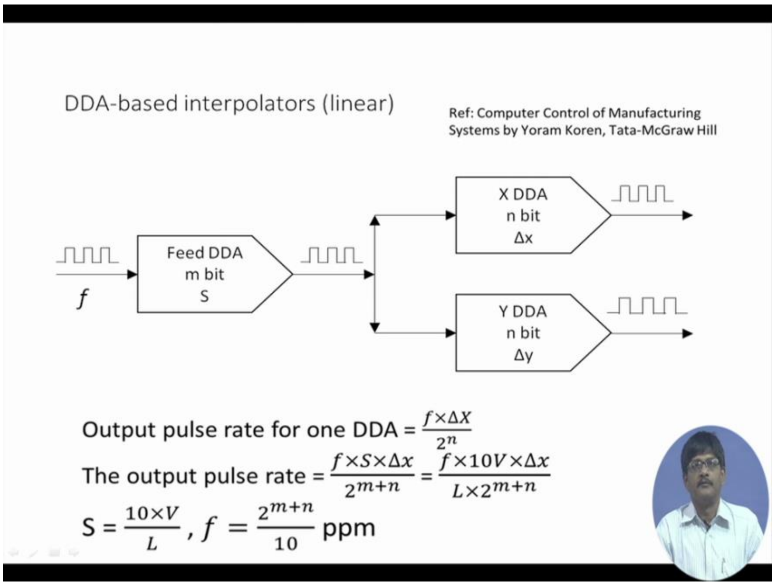
So we can solve the problem this way, we 1st understand that the basic length unit must be equal to leads screw pitch divided by the encoder number of holes, which is 20 microns 0.02 millimeters. Amount of movement along X, we found out just now, it is 30 millimeters, movement along Y = 40 millimeters, this also we have found out just now. Displacement ratio is equal to velocity ratio for straight-line movement, what does this mean? It means that whenever motion is taking along a line, a part of it along X, a part of it along Y, in that case the velocity along X and velocity along Y will be having the same ratio as the displacement along X and displacement along Y in case of linear motion.

So as we are having linear interpolation here, so V x / V y = Δx/Δy=3/4 and we also know since the feed velocity is 100 millimeters per minute therefore, the resultant velocity along the cutter path must be $\sqrt{V_x^2 + V_y^2}$=100. From this, we can easily find out the value of V x = 60 millimeters per minute, so once V x has been found out we can find out that sorry when V x has been found out from the basic length unit which has been calculated, we can find out the rest of the terms required.

For example, when the table is moving at 60 millimeters per minute along the X axis, the encoder must be sending 60 divided by 0.02 pulses per minute, why is this so, because 60 millimeters per minute to the encoder is not exactly 60, but it is equal to the number of basic length unit it has to generate in the corresponding time, so let us see 60 divided by 0.02 must be equal to 3000 pulses per minute. So in order to develop a speed of 60 millimeters per minute or rather when the table develops the 60 millimeters per minute, encoder sends 3000 pulses per minute, so now we understand that the encoder pulse rate is 3000 pulses per minute.

So at steady state for linear velocity, the encoder pulse rate must be equal to the interpolator pulses rate, what does this mean? We already established that when we are having linear velocity, a steady state has been reached and at that time interpolator pulses becomes equal to the encoder pulse rate. Since encoder pulse rate has been found out to be 3000, interpolator pulse rate also must be 3000, and so answer is interpolator pulse rate must be 3000 pulses per minute. And how many pulses are being pass through at the AND gate, it must be equal to it must be corresponding to 30 millimeters because 30 millimeters is the movement along X, divide 30 millimeters by 0.02 and you will get 1500 pulses, so 1500 pulses are being sent through so this is the answer to that question.

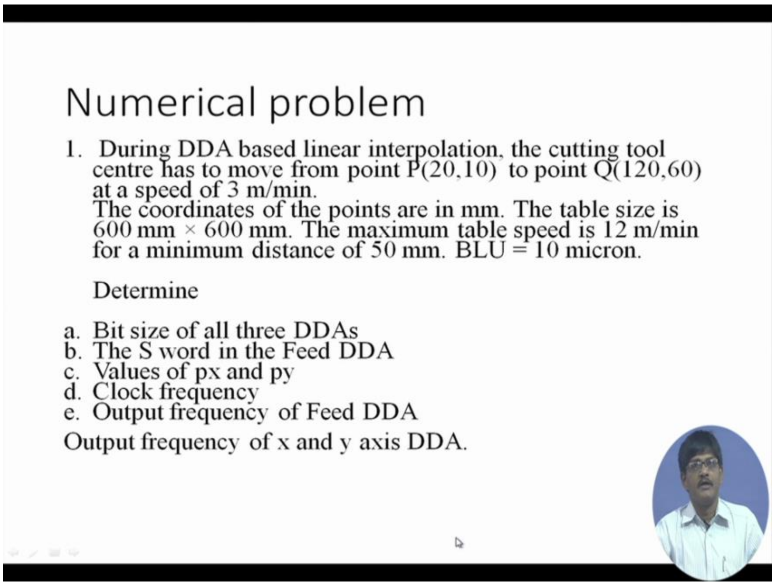
(Refer Slide Time: 11:39)



Now let us see a DDA based interpolator, in the DDA based interpolator as we have discussed previously there is an input pulse rate *f*, it enters a feed DDA which is having m number of bits and inside this m bit counter a number S is stored, so S has to be smaller in size compared to $2^m - 1$ that that can be the maximum size of S. The output pulse rate of any such DDA is always equal to *f* multiplied by Δ x which is inside that particular counter divided by 2^n if n be the number of bits of that DDA. So the pulse rate output of the feed DDA goes to X DDA and Y DDA and they send out their respective pulse rates to the X motor and the Y motor, this is the basic structure of the linear interpolator.

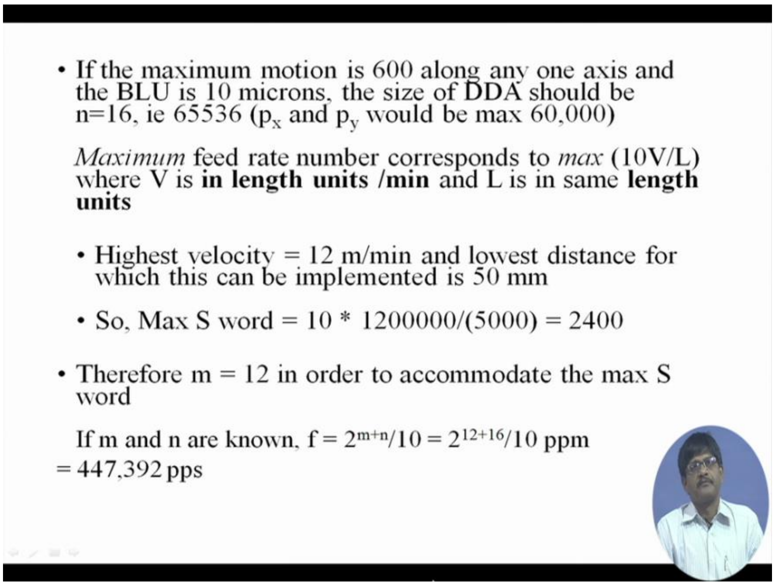
And output pulse rate is ultimately found out to be the input pulse frequency multiplied by n×V, V is the programmed feed rate along the cutter path into Δ x, Δ x is the incremental X motion divided by L, L is the length segment of the path and multiplied by 2^{m+n}, this is the pulse rate out of the X DDA, so let us and S also has an expression, 10 × V / L, where these things we have already defined and F has a value of 2^{m+n} / 10, so let us take a numerical problem.

(Refer Slide Time: 13:39)



During DDA based linear interpolation, the cutting tool Centre has to move from point (20, 10) to the point (120, 60) at a speed of 3 meters per minute. The coordinates of the points are in millimeters, the table size is 600 × 600 millimeters, the maximum table speed is 12 meters per minute for a minimum distance of 50 millimeters and the basic length unit is given to be 10 microns. Find out the bit sizes of all 3 DDAs, the S word in the feed DDA that means the S word which has to be put in this particular case, the values of px and py, the clock frequency and the output frequency of feed DDA and of course the output frequency of x and y axis DDA.

(Refer Slide Time: 14:45)



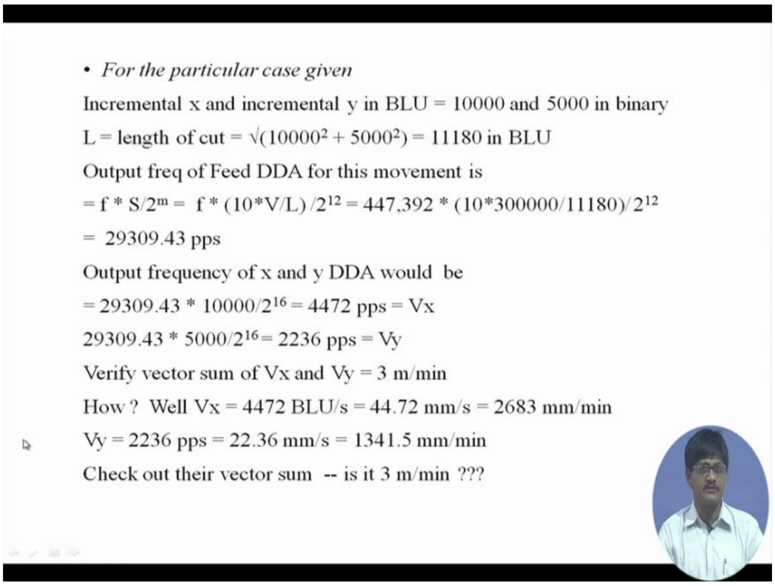
So first of all let us see the bit sizes of all three DDAs. So we understand that 600 millimeters is the table size on either X or Y direction, if this be so then if the basic length unit is 10 microns, in that case we can easily find out what will be the maximum size of Δ x that means maximum possible incremental motion. So if we suppose that the maximum possible incremental motion is equal to the length of the table itself, which is 600 millimeters therefore, 600 divided by basic length unit is it must be equal to those many number of bits or the size of the size of the number which can represent which can be the maximum Δ X or incremental X motion.

So is 600 millimeters is our motion, then 600 ×100 is the maximum number which has to be accommodated in the X feed DDA X DDA okay, so let me read it. 60,000 being equal to 600 divided by 10 microns is the maximum number which has to be accommodated in the X DDA counter. Now X DDA counter is of n bits, so the maximum number it can accommodate is 2 to the power n - 1, so simply by calculation we can find out if n = 16, then $2^n - 1$ is 65536 and this is the smallest counter which can accommodate 60,000, so we come to the conclusion that the size of the counter inside X and Y DDA must be 16 each, 16 bit counter inside X and Y DDA.

In the same way, we understand that the feed DDA which has S word inside which is of m bit size, it can contain or rather it has to contain the maximum possible feed rate number or S word. In that case, the maximum possible number is given by the condition given in the problem, the condition is the maximum table feed or rather table speed is 12 meters per minute for a minimum distance of 50 millimeters, so we understand that if we take a a table size sorry table speed of 12 divided by the corresponding length of 50 millimeters, we will be able to compute the maximum value of the S word.

So the highest velocity is 12 meters per minute and lowest distance is 50 millimeters therefore, maximum S word is 10 × V / L, so this thing is expressed in length units per minute, so this is equal to 2400. So in order to accommodate 2400, which is the maximum possible S word, we have to have m bits of sorry m = 12, so 2^{12} - 1 will be able to accommodate 2400. Having understood that, we understand that m = 12 and n = 16 so that the input clock frequency which is equal to 2^{m+n} divided by 10 pulses per minute, we can easily compute it to be 2^{28} divided by 10 pulses per minute = 447392 pulses per second. So f has been computed as we are already as we have already calculated the values of m and n.

(Refer Slide Time: 19:34)



Coming to the 2nd part of the problem, for the particular case that has been given, incremental x and incremental y in basic length units, they are 10,000 and 5000 respectively, how did we find this out? We found out the initial point and the final point, the differences Δ x and Δ y divided it by the basic length unit and obtained these 2 values. Basic length unit incidentally just to remind you was 10 microns, so now the length of cut okay, the length of cut must be the square root of their of the sum of squares of these 2 and therefore, it is equal to 11180 in basic length unit okay. So why are we finding out the length of cut because V/L has to be calculated for this particular case.

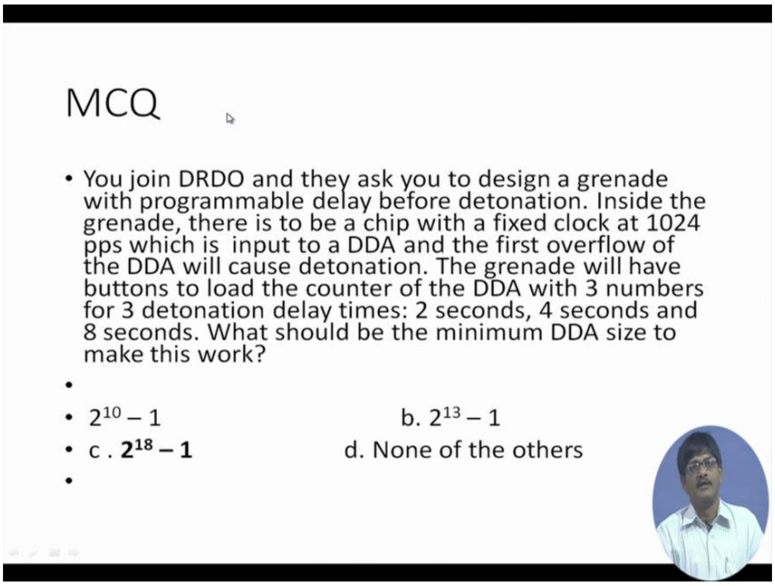
So output frequency of feed DDA for this movement, so what is this feed DDA? That is the first DDA which was existing, let us have a quick look this is the feed DDA, we will find out the output frequency of this one then multiply this by the corresponding Δ x value and that way we will get these two frequencies. So output frequency of the feed DDA for this movement is *f*, using the new know same only one formula is there, F × S divided by 2^m since we know all the values now, we put 447392 as the value of F, we put 10×V = 10× 300,000 that means velocity is 3 meters per minute given for a particular case that is divided by the basic length unit.

So basic length unit per minute divided by length expressed in the same length units that means basic length units divided by 2^m, which is 2^{12} and that way we obtain 29309.43 pps we are having this in pps because *f* is in pps (pulses per second). So having obtained the output frequency of feed DDA, we can easily obtain the frequencies of x and y DDA. How is this so, because feed DDA multiplied again by the same form that is *f* × X / 2^m, so we have feed DDA frequency multiplied by Δ x divided by 2 to the power m where m = 16 will give us 4472 pulses per second, this is V x.

Once again, feed DDA output frequency multiplied by the Δ y value divided by 2^{16} 16 gives us 2236 pulses per second, this is V y. Now the vector sum of V x and V y is to be 3 meters per minute, so if V x = 4472 basic length unit per second, it is correspondingly equal to 44.72 millimeters per second = 2683 millimeters per minute. In a similar manner, if we compute V y, it will come out to be 1341.5 millimeters per minute and therefore, if we combine this these 2 vectorially, their result will come out to be 3 meters per minute.

So therefore we have found out first of all m, we have found out n, n is 16, m is 12, *f* has been found out as 447392 pulses per second, the output frequency of the feed DDA has been found out to be 29309.43 pulses per second and V x and V y frequencies have also been found out, okay. Let us take one multiple-choice question.

(Refer Slide Time: 23:58)

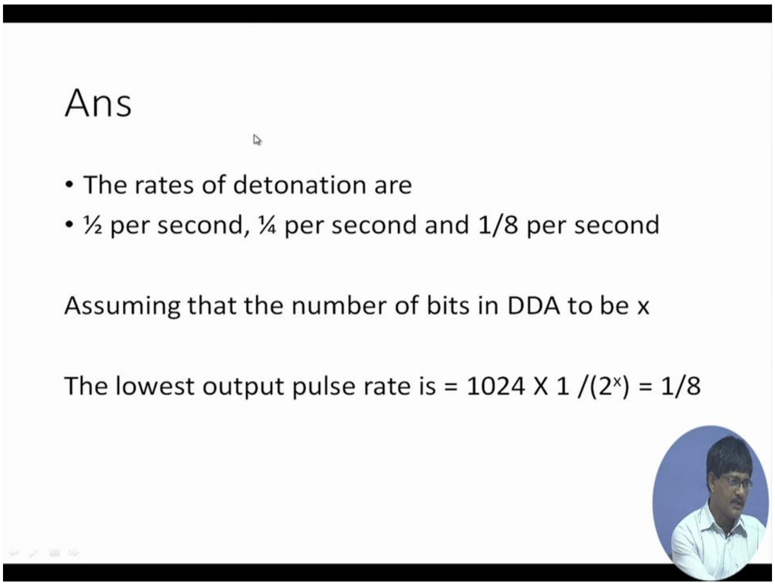


In this multiple-choice choice question what is given, you are joining DRDO which we already know to be Defense Research Development Organisation and they ask you to design a grenade with programmable delay before detonation. grenades, when you take out some pin from that, after that a definite time is permitted beyond which an explosion or detonation occurs. So here we are talking about a grenade where the detonation time or the donation delay is programmable. Inside the grenade, there is to be a chip with a fixed clock at 1024 pulses per second, which is input to a DDA.

So all these things are inside the grenade because DDAs and clock small clocks, et cetera they are miniaturized and they can easily fit into grenade, so which is input to a DDA and the first overflow of the DDA will cause detonation, so it is simply like this. I have a clock, I have a DDA, I am putting in some number and the first overflow pulse which comes out, first output pulse it is going to cause detonation. The grenade will have buttons to load the counter of DDA with 3 numbers for 3 detonation delay times; 2 seconds, 4 seconds and 8 seconds, what should be the minimum DDA size to make this work?

So we know *f*, we do not know the number of bits inside the counter of the DDA and we know the required pulse rate output at the detonation side. So the answers are 2^{10} - 1, so DDA size we know it is 2^n - 1, so that way we have given it in that format. 2^{10} - 1, 2^{13}- 1 and 2^{18} - 1 and none of the others of course, so let us try this out.

(Refer Slide Time: 26:27)

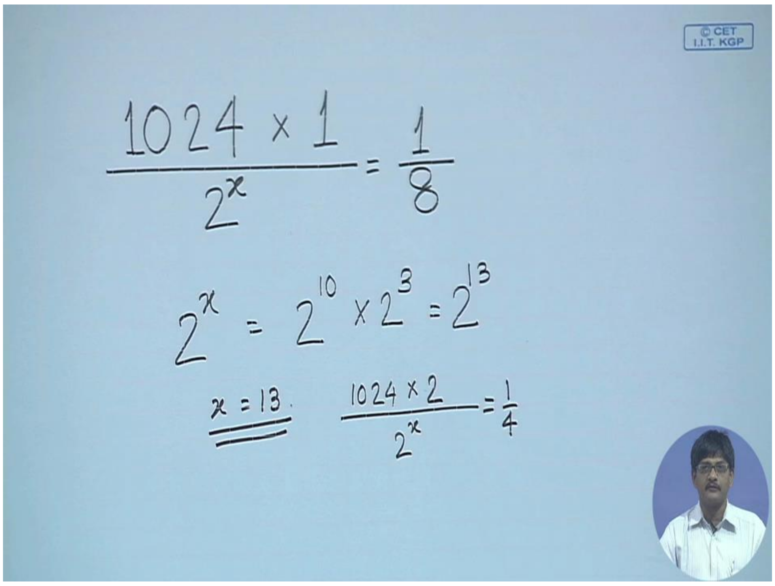


So the rate at which the detonations are supposed to take place that is half per second because if 2 is the time delay for detonation, that means the first detonation is going to take place after 2 seconds have passed and therefore, if one detonation takes place in 2 seconds therefore, per second I am having half a detonation this is the rate of output pulses required for this particular case. In the same way, if 8 seconds are supposed to pass before the detonation therefore, 1 by 8 detonation per second is taking place. So we have the frequency of detonation to be half, one fourth and one eighth per second.

So let us assume the number of bits in DDA to be x, so what does this mean that the counter size in the DDA is equal to x bits, so the maximum number that it can contain is equal to 2^x x - 1. So now let us look at the lowest output pulse rate that can be obtained from this DDA, what do we mean by lowest output pulse rate? 1024 is fixed, the frequency then hardware size we are already assuming to be of x bit, so 2^x denominator is fixed and therefore, the x which gets multiplied as the number contained in the counter, if it is the lowest possible value x inside the counter can only take up integer values that means its minimum value can be 1, if it is 0 then there is no point having that that DDA.

So minimum value is 1 and then you can take other positive integer values. So, if we are planning to get the slowest detonation rate at one eighth per second, let us equate 1024 divided by $2^x \times 1$ that means I will put a 1 bit there inside, so let us work out and see how much that comes out to be.

(Refer Slide Time: 29:04)



We have here 1024 is the frequency multiplied by only 1 here divided by 2^x equal to the frequency that we want. If I put the smallest number I should get the lowest frequency. So that way, 2^x becomes equal to… 1024 multiplied by 2^3 3, since powers are added I can write 2^{13} and therefore, x becomes equal to 13. So if you have a the number of bits inside this counter to be 13, then we can definitely ensure that this is going to have detonation times very easily by putting 1024 divided by 2^x ×2, this is going to give us develop this, therefore we are going to get one fourth.

So on and so forth, if you put 3 were going to have 1, so this way we can make this particular grenade work okay, thank you very much thank you.