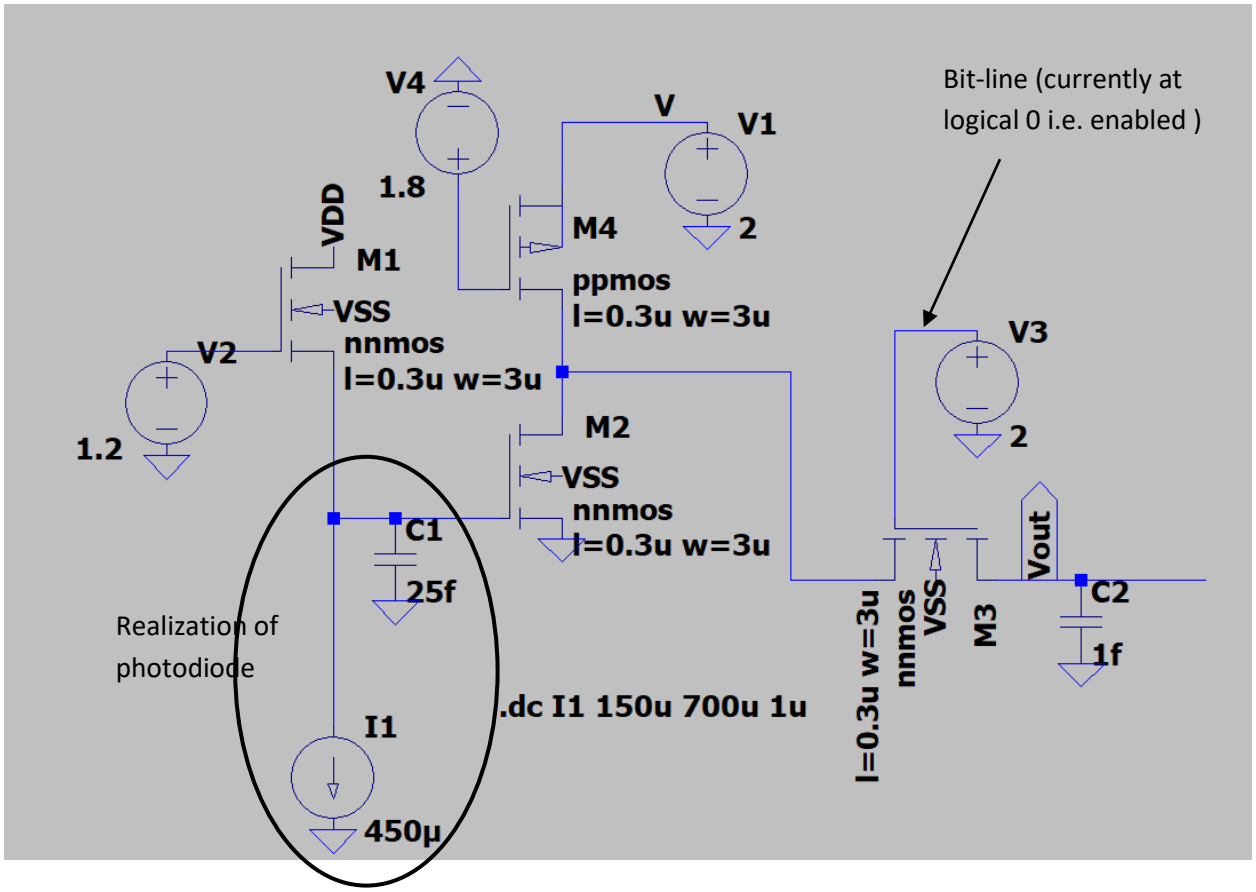
# Verification of 16*16 Active Pixel Sensor using LtSpice and Matlab

**Abstruct :** Here we will verify the conventional **Active Pixel Sensor (APS)** circuit with a 16*16 pixel image and also save the data in form of voltage (in a matrix form). Finally by applying some linear mapping we will try to reconstruct the image in **Matlab** and note down the **PSNR** .A high PSNR will indicate that this circuit can be used for capturing photo perfectly .

**Introduction :** Active Pixel Sensor is a widely used technology for capturing image digitally. In the actual circuit there has to be a photo diode ,where, if light falls a certain amount of current is generated and by virtue of the circuit the current is converted into a well defined voltage which is then digitized and saved. There is one **Bit-line** which is used to serially push the data of each row .If the **Bit-line** is high(1) the output voltage can come to the output .Its like an enable switch .A large number of pixels are arranged in a matrix fashion to capture 2-D image .While digitizing the image Bit-line comes into the picture. Usually a NMOS (or PMOS) is used as an switch in this case which is enabled by logical 1 (or logical 0) .Each pixel has this kind of structure. During digitization one whole row at a time gets the enable signal i.e. a logical 1 (or logical 0). Finally when all the rows are over ,we can say that the full image is saved.

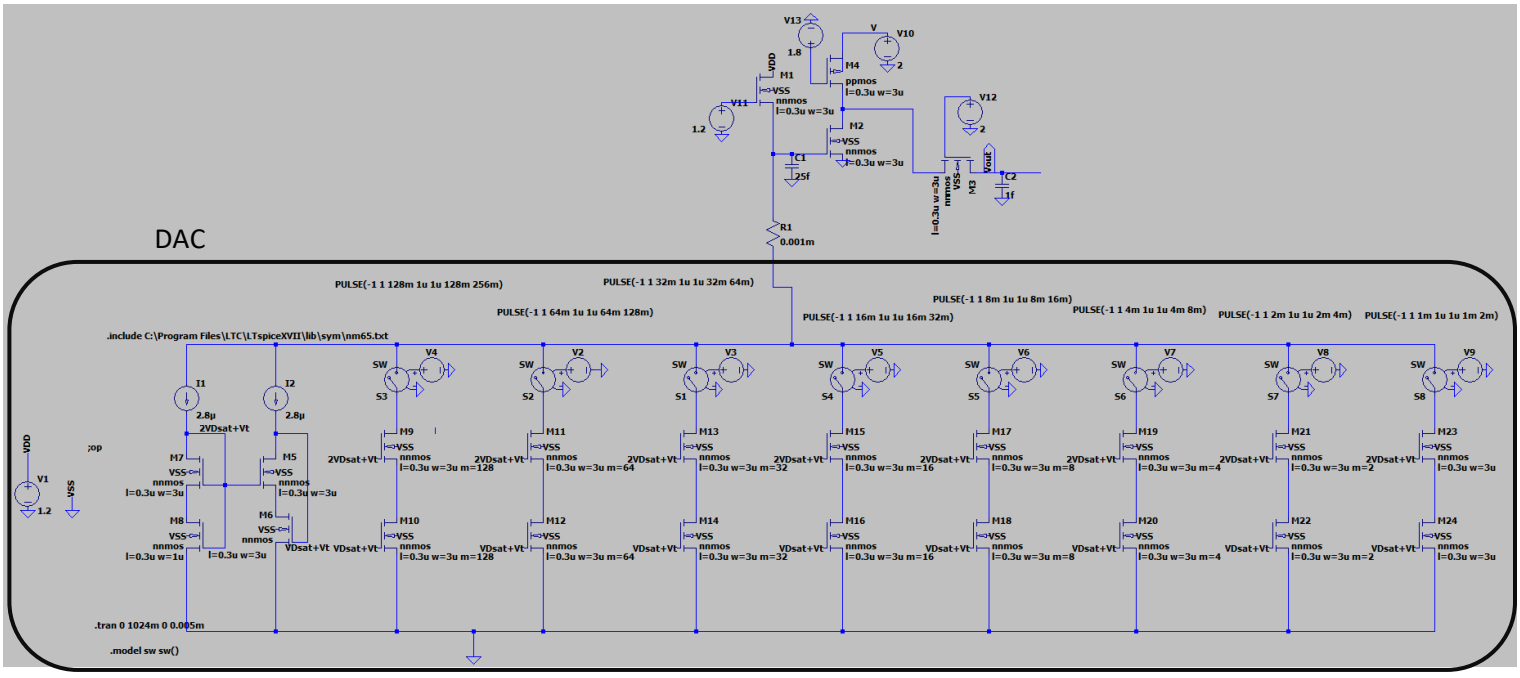
**Realization of one unit (pixel cell) of an APS :**



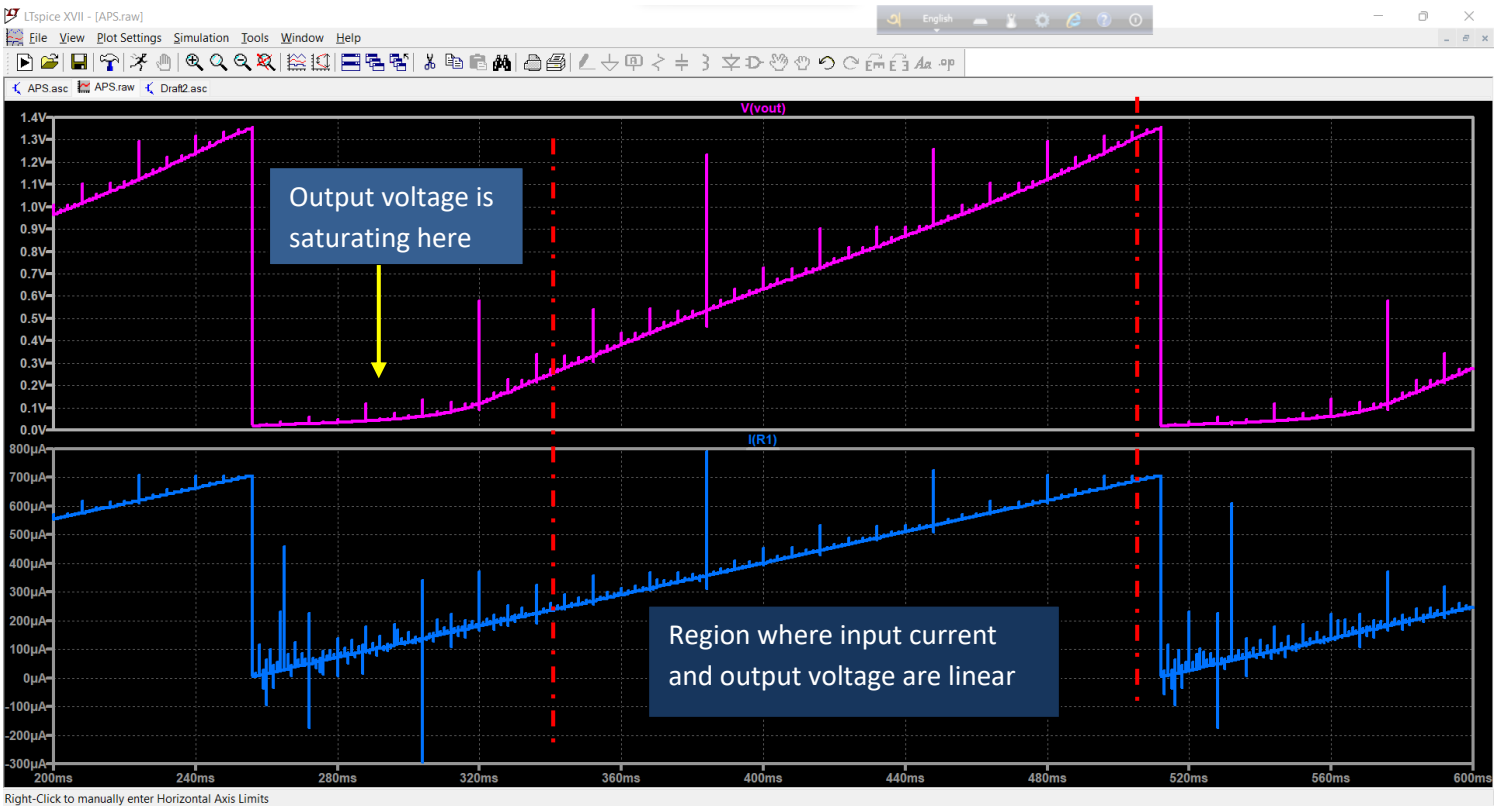
The photo diode is realized with an ideal current source and a capacitor .Here capacitor won't allow to change the voltage and current source is for realizing the current generated in photo diode. Here depending on the current the floating node will generate a necessary voltage to bias the NMOS (M1) properly. This generated voltage will be amplified by the CS- stage amplifier (with an active load) next to it and then finally is bit- line gets a high input we can have a well defined voltage at the output.

**Flow of the verification :**

1. At first we have to find the range of the current where the NMOS M1 is in Triode i.e. the current is linearly varying with VGS of M1 which is to ensure that by applying simple linear algebra we can get back the image. For this ,first we will drive the pixel unit with a **low-voltage Sooch current mirror based DAC**. This DAC will basically act as a current source for the NMOS M1 .Also the voltage controlled switches present at the DAC will be used for digital input .It is a 8-bit DAC .We have to gradually increase the digital input from 00000000 to 11111111 and corresponding to that we have to record the current and voltage values at 'Vout' .We will do this by transient simulation .From the graphs we can get the idea that in which region of values of current the voltage comes as a linear function of current such that there is a linear one-to-one relation between current generated by photo diode and the output voltage that is to be stored.



## Simulation result



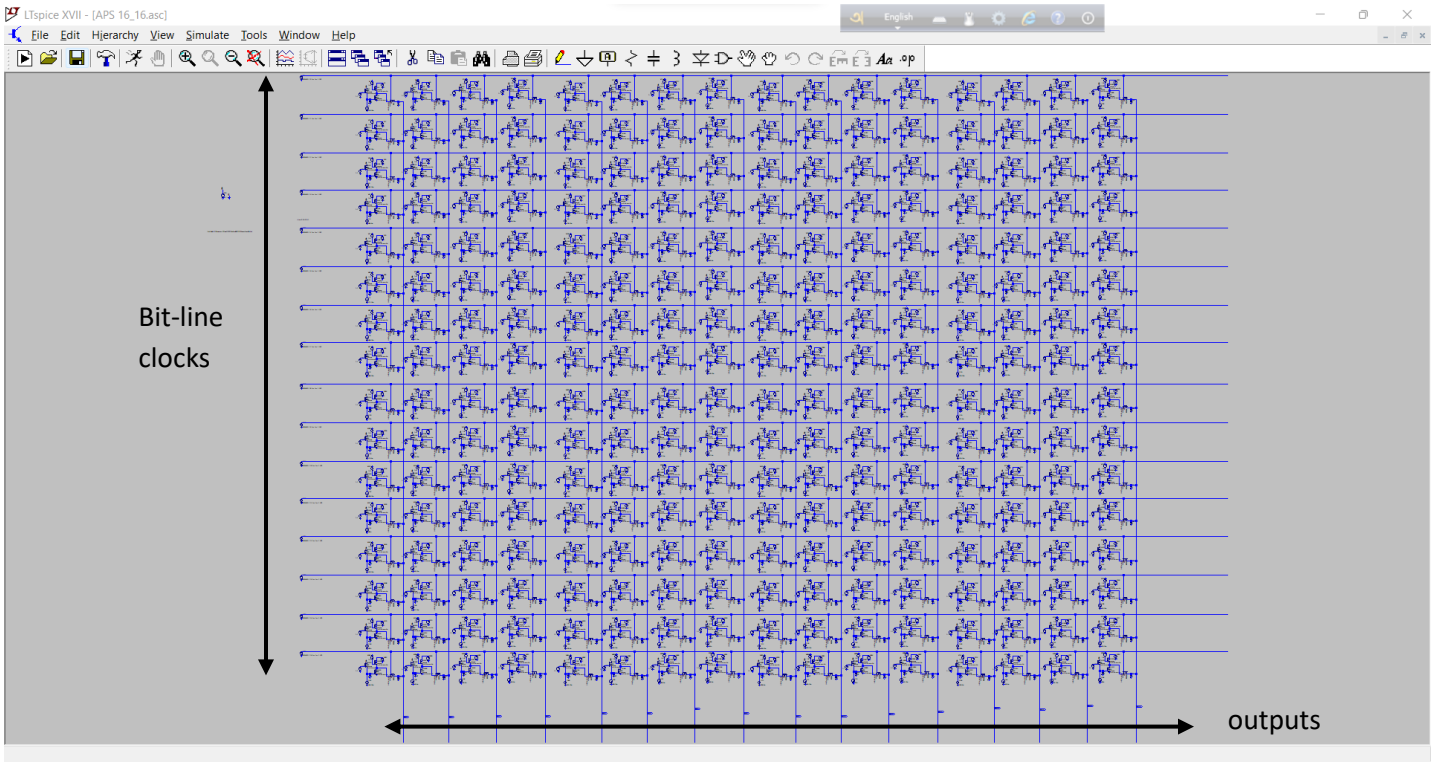

2. We have to measure the boundary current and voltage of this region now. The values are : **218 uA (215 mV)** and **665 uA (1.245V) .**
3. Now we have downloaded an image of 16*16 pixels and read that in Matlab .But in matlab the image is usually taken as an **unit8** type matrix by default which actually means 8-bit quantized image .Now from this we have to generate the sets of input currents which needs to be put as the value of the ideal current sources of each pixel cell of the 16*16 APS schematic .
   So here we have to do a linear mapping from the values **(0 to 255) to (218uA to 665uA)** .

## Matlab Code

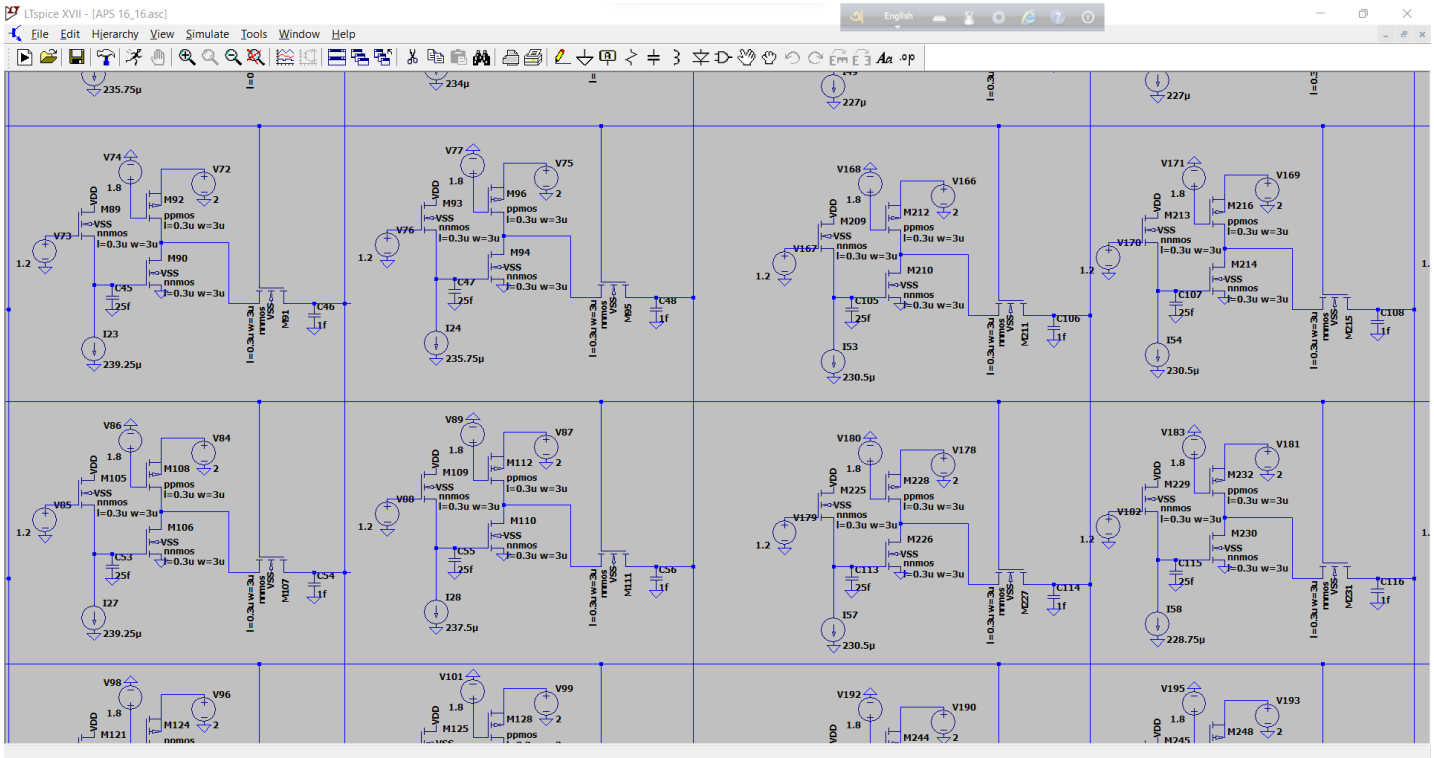
```
im=imread('APS.jpg');
im=double(im);
im=(im+124.7142857)*1.75;  % Linear mapping
dlmwrite('imagetocurrent.txt',im,'delimiter','\t','newline','pc');
```

4. Now we have to make the final 16*16 pixel APS schematic .Here we have to take care of the clocks provided to the Bit-lines also. It should be such that at a time only one row should get high voltage (logical 1) to pass the output.

## Schematic (16*16 APS)

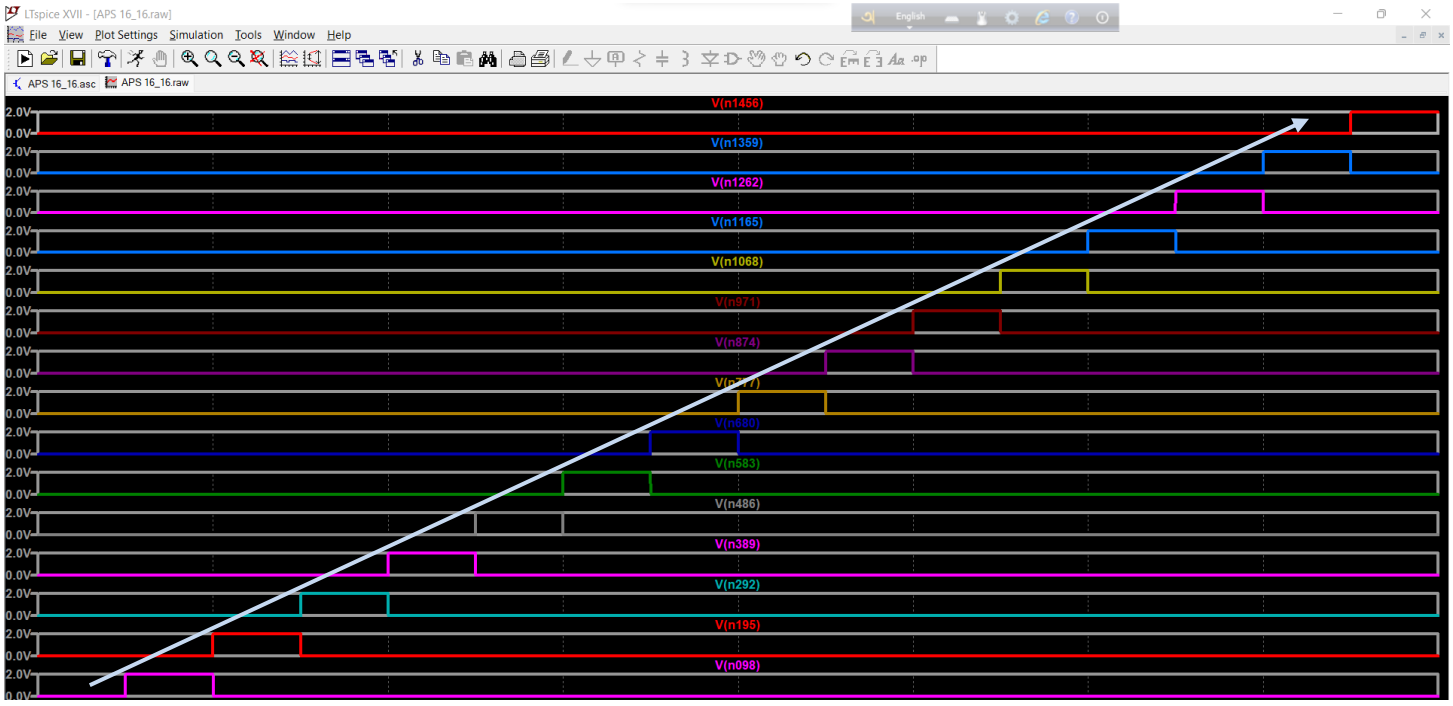



**A closer view**



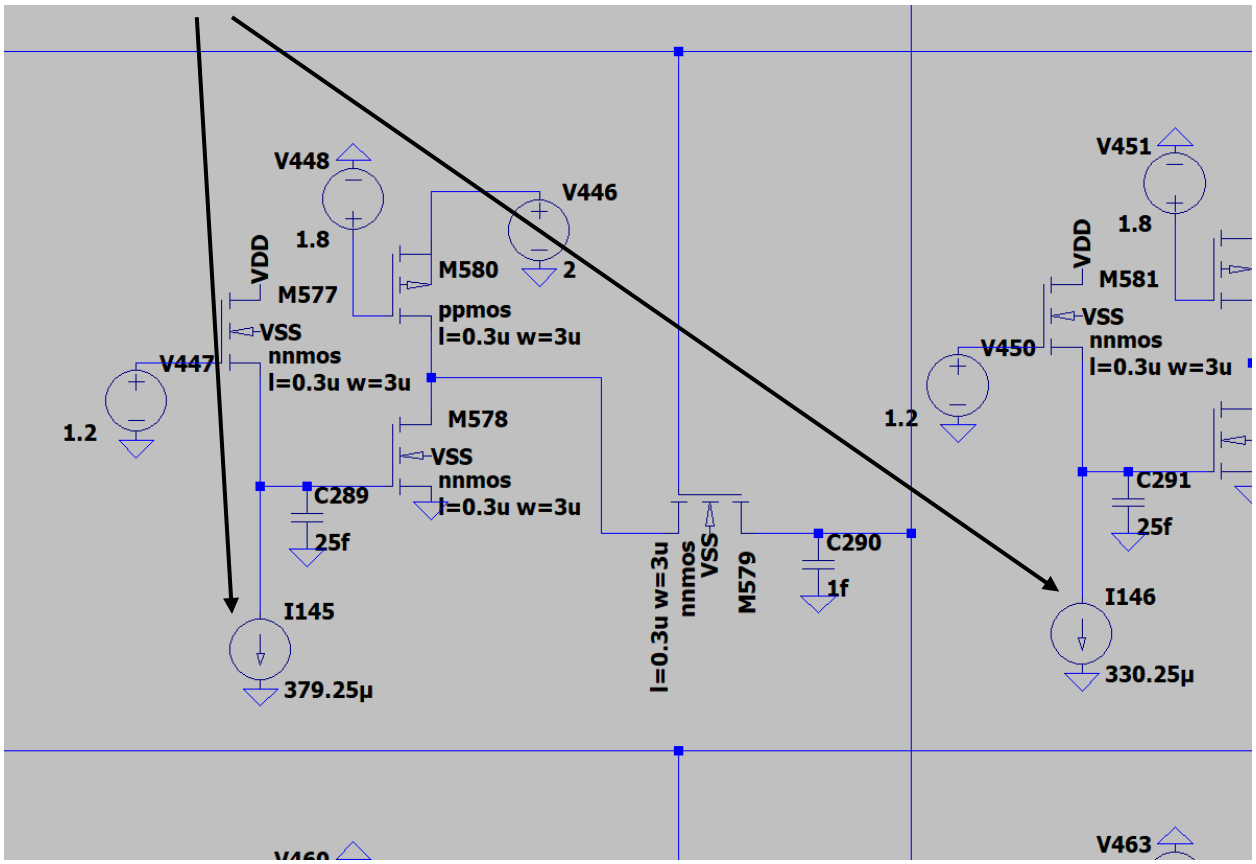

**The Bit-line voltages are provided by 16 individual ideal Pulse voltage sources with different delays.**

## Generated clocks for Bit-lines




**So in each 1 second an entire row will be read.**

5. Then the 256 current values were filled manually in the circuit to realize photo diode effect.



6. Then finally the output of 16 seconds are collected and imported to a .txt file where it is in matrix format. Finally we again switched to Matlab and read that matrix in Matlab. But for plotting we again have to linearly map the values **from (215mV to 1.245V) to (0 to 255)** .After the mapping we have extracted it as an image and computed the PSNR.

**Matlab Code**

```
im=readmatrix("APS 16_16.txt");
im=(im-0.213547)*247.2240616;                % Linear mapping
im=round(im);
im(im>255)=255;
im(im<0)=0;
im = uint8(reshape(im, size(im, 1), [], 1));
imshow(im);
```

**Original Image** **Reconstructed Image**

Calculated PSNR = 49.54 dB

**Conclution :** Here we have successfully verified that our pixel cell works perfectly .We have also seen that by carefully controlling the clocks of the Bit-lines how we can capture an image in real time. We have also got a pretty good amount of PSNR. Ideally it should be infinity but in all practical cases due to limited bits in digitizer it is not infinity but a large number.

**********