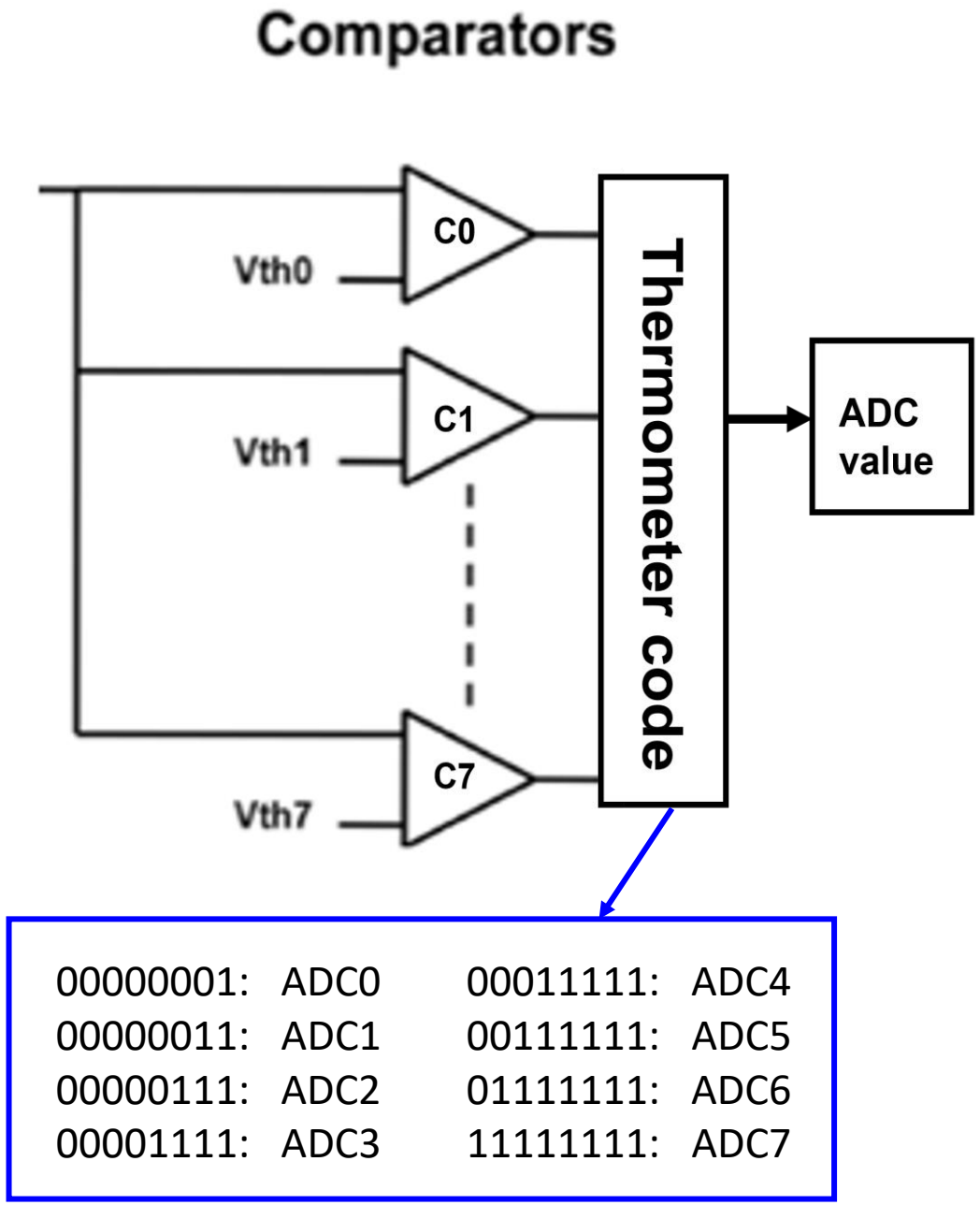


Double counting issue

- **ADC0**:
  - vth0 < amp <vth1
  - C0 switch ON.
- **ADC1**:
  - vth1 < amp <vth2
  - C0 and C1 both ON
- Above two condition are **mutually exclusive**.
- The test pulse sample produce ADC0 and ADC1 are thus not overlapped.
  - i.e. not double counting between the two samples

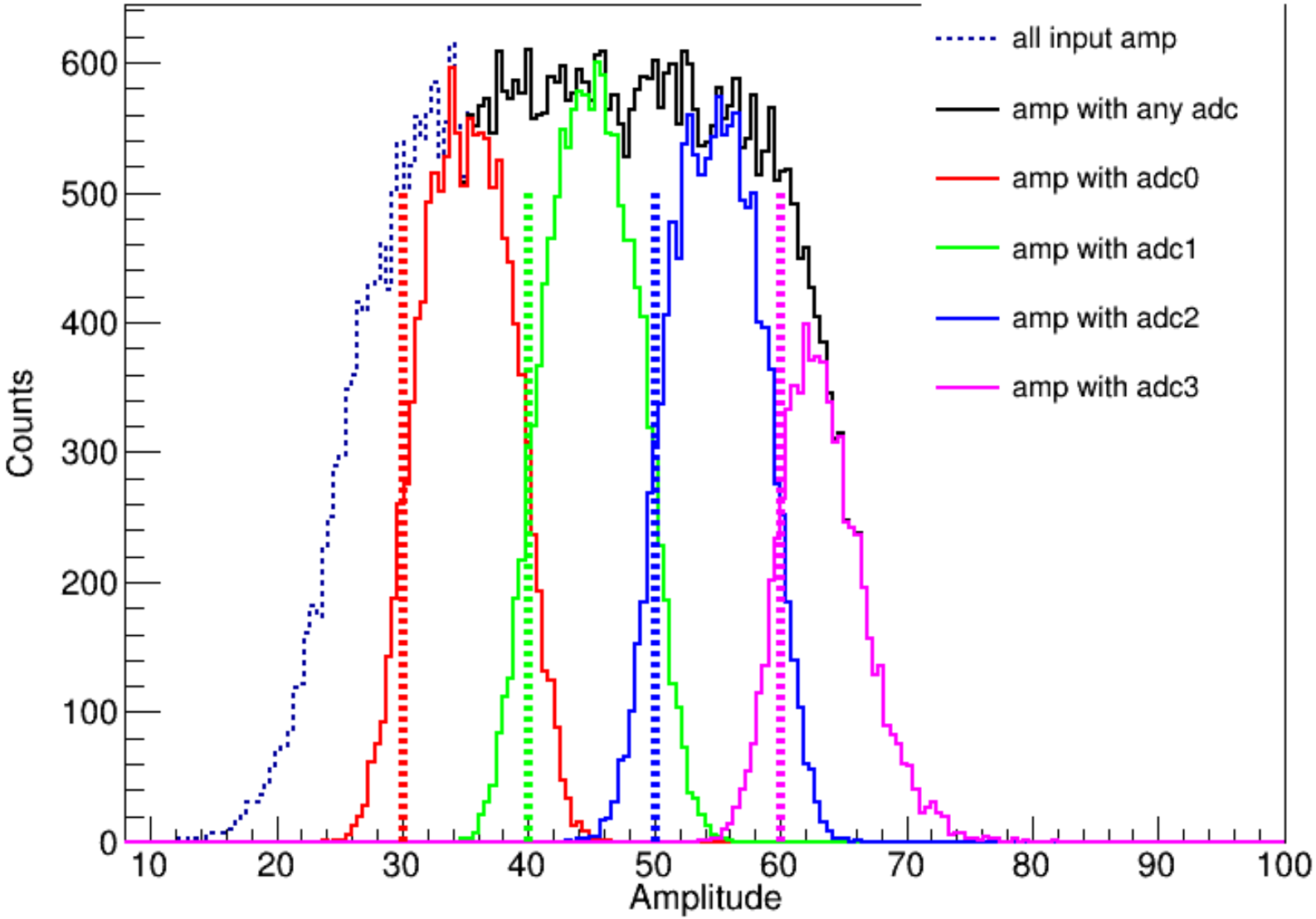
A Toy model: 2-bit ADC

```
const int n_pulse = 50000;  // number of test pulse
const int n_adc = 4;  //.. number of ADCs

float amp_mean[n_adc] = {30, 40, 50, 60};  // Gaussian mean of input pulse amplitude
float amp_width[n_adc] = {5, 5, 5, 5};   // Gaussian width of input pulse amplitude

float thres_mean[n_adc] = {30, 40, 50, 60};  // Gaussian mean of threshod of adc0, adc1, adc2, adc3
float thres_width[n_adc] = {2, 2, 2, 2};  // Gaussian width of the threshold of adc0, adc1, adc2, adc3

for(int i = 0; i<n_pulse; i++) {
  int idx = int(gRandom->Rndm()/0.25); // each amp_mean has 25% probility to generator pulse

  // amplitude of input test pulse
  float amp = gRandom->Gaus(amp_mean[idx], amp_width[idx]);

  // threshold for each comparator
  float thres[n_adc] = {0};
  for(int it = 0; it<n_adc; it++)
    thres[it] = gRandom->Gaus(thres_mean[it], thres_width[it]);

  // comparator 0-3 initially OFF
  int c0 = 0;
  int c1 = 0;
  int c2 = 0;
  int c3 = 0;
  // now determind the ADC value of this test pulse
```

Note: code posted in mattermost INTT channel.

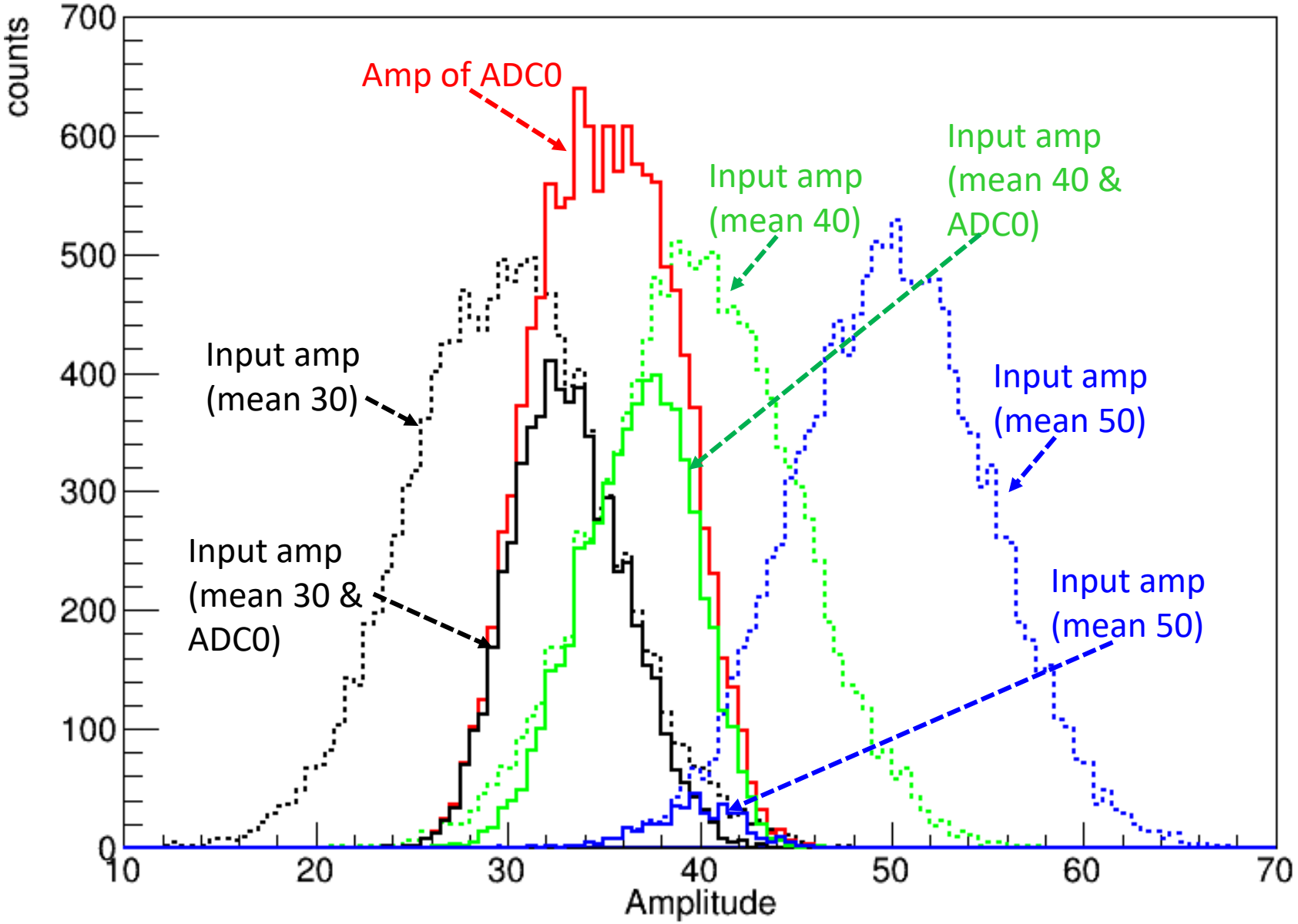
```
// now determind the ADC value of this test pulse
float adc = -1;  // initial value
if(amp >= thres[0] && amp < thres[1]) {   // comparator C0 ON
  adc = 0;  // this test pulse output adc0
  c0=1;
} else if(amp >= thres[1] && amp <thres[2])  {  // comparator C0 and C1 ON
  adc = 1;  // this test pulse output adc1
  c0=1;
  c1=1;
} else if(amp >= thres[2] && amp <thres[3])  {// comparator C0,C1 and C2 ON
  adc = 2;  // this test pulse output adc2
  c0=1;
  c1=1;
  c2=1;
} else if(amp >= thres[3]) {// comparator C0,C1,C2 and C3 ON
  c0=1;
  c1=1;
  c2=1;
  c3=1;
  adc = 3; // this test pulse output adc3
}
```

Amplitude distribution from a 2-bit ADC



The vertical dashed line corresponds to the threshold for adc0, adc1, adc2, adc3

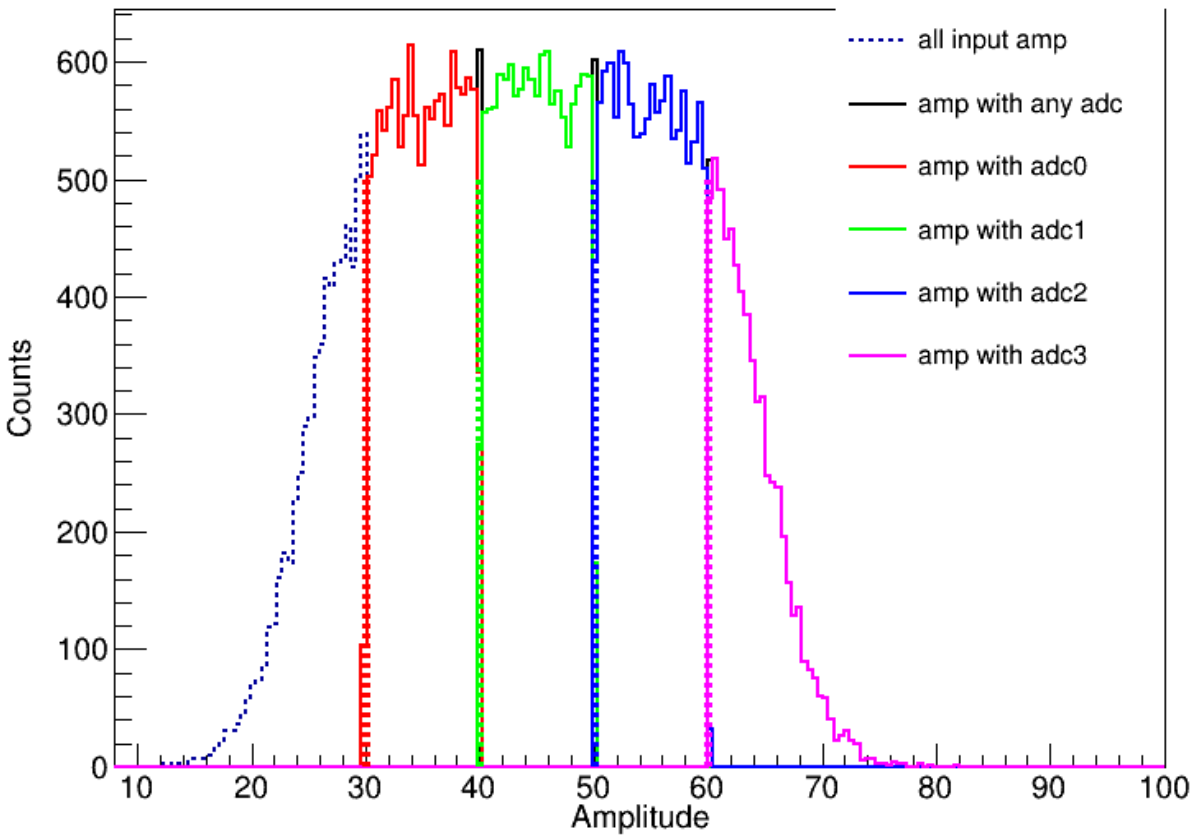
Why the amp distribution of each ADC are Gaussian-Like



Amp of ADC0 =

- Input amp (mean 30 & ADC0) +
- Input amp (mean 40 & ADC0) +
- Input amp (mean 50 & ADC0)

When the threshold noise is zero



Color coding same as in the last slides