Requirements from peripheral logic

1. For a double column (12.8mm\*0.05mm) readout, how many hits (or pixels ) in maximal need to be read in 3us (the period of trigger latency) ?

--typically 1 cluster, i.e., 4 hitted pixels

--When I decide the FIFO volume, I hope to evaluate the maximal hits per double column. At least, it cover the possibility of 99.99999% (the actual possibility should come from CEPC requirements).

--At the moment, I suppose that there are less than 12 hit pixels in 10 us (considering the largest trigger latency of 6.4us).

1. How long is the acceptable dead time for the pixel readout?

-- 2 clks for 1 pixel. i.e., If 4 hitted pixels in 1 cluster, about 200ns dead time are expected.

-- I prefer to process more columns together in order to reduce the area and the power consumption of the peripheral logic. So I want to know whether the dead time of 300ns, 400ns, 600ns, even more are acceptable.

1. Are CLK\_40MHZ and CLK\_160MHZ synchronous? How about the duty cycle and the jitter for CLK\_40MHZ and CLK\_160MHZ, respectively?

-- 40MHz and 160MHz clk are synchronized.

-- Is it so important? 40MHz is from backend electronics and I can only say it is in a normal range at current stage. You can propose your requirement for the jitter if necessary (and if the backend people can implemented).

160MHz is only for the output serializer.

--In the triggerless mode, the data rate is 120 MHz. I will use 160 MHz clk for data output. All the other design use 40 MHz clk.

--It is important for the design synthesis. We use both of the positive edge and negative edge of 40 MHz clock, so we need to know the duty cycle. The duty cycle and the jitter will be considered for the design synthesis. Different parameters will result in different designs. At the moment, I ask our designer to apply the clock duty cycle of 50% and the jitter of 200ps. When backend people provide the real parameter, the design synthesis and the following work should be redone.

1. In-Pixel Digital
2. As shown in Fig,1, TDA=?, TDFOR=? (TDA includes the delay of READ and the delay of Address encoding.)



Fig.1 Timing sequence of double column readout.

--Tianya please give the estimated value

1. Please give the addresses in a double column like Fig.2.



Fig.2 Example of the address encoding in a double column

--Comments: What is the point and what is the must by doing so? The address encoder is not finalized now, and if taking account for the pixel 7 as an example, if it is encoded within the single column, the address is only 3 bits: 111，but if encoded in the up given way, the address is 4 bits: 0111. It means the column bit 0/1 is duplicatedly added into pixel, which will add extra area cost.

--The length of the encoded address will be same. For two single columns, I will add an extra bit to identify the columns when the addresses are written into FIFO.

--The considerations of data processing with double column: (1) Refer to FE-I3 and ALPIDE, they adopt double column, (2) the two columns can share the readout circuits, memory and so on.

--I presented this address encoding in our meeting on 6th Aug when we discuss the design interface. Our present design is based on the double column. It will be too late to restart the peripheral design after the implementation of address encoding. So we must decide now.

--I concern the detail address encoding for data compression. The data compression function may reduce the access times of memory and the data rate, then the power consumption can be reduced.