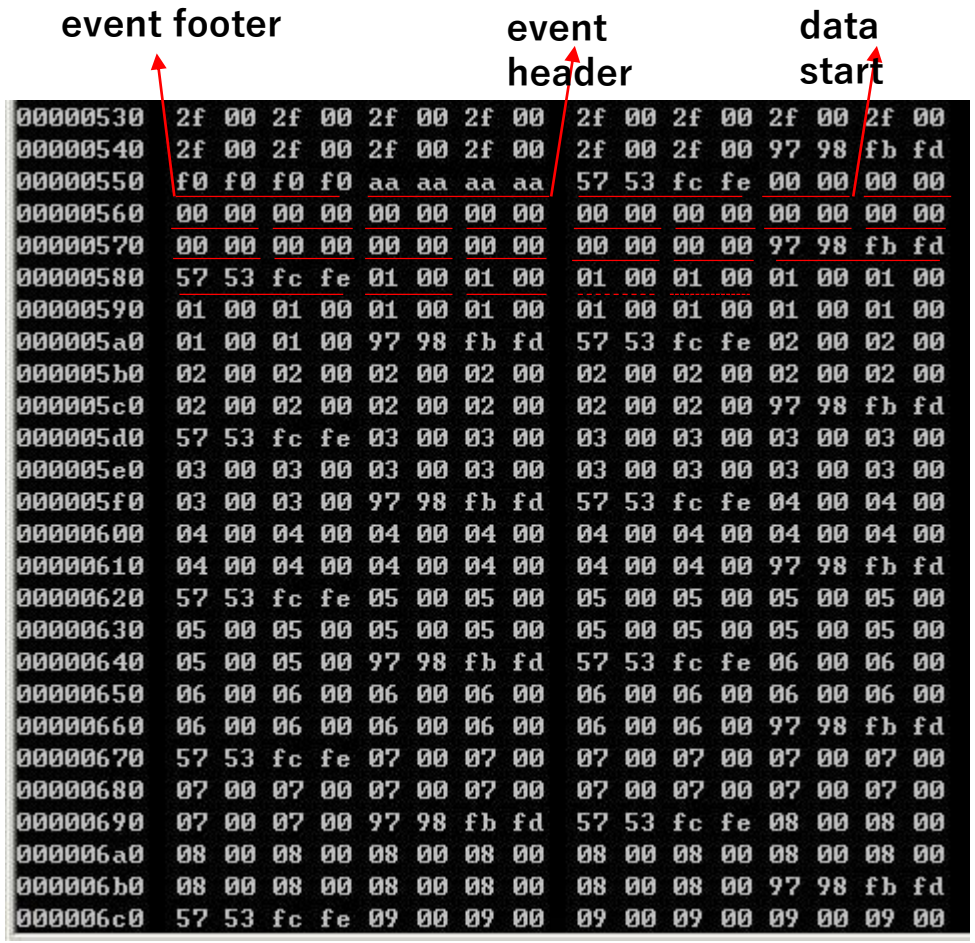
Contents	Value	row number
Event Header	0xaaaa_aaaa	
Row 01 header	0x5753_fcfe → **Changed from right figure as below** => 16it (0x5357) + 16bit(“1111”(4bit) + [event_type: 2bit =“10”(fixed)] + [row count: 6bit] + [internal event counter: 4bit])	row 01
pixe01/pixel02	0x0000_0000	row 01
pixe03/pixel04	0x0000_0000	row 01
pixe05/pixel06	0x0000_0000	row 01
pixe07/pixel08	0x0000_0000	row 01
pixe09/pixel10	0x0000_0000	row 01
pixe11/pixel12	0x0000_0000	row 01
pixe13/pixel14	0x0000_0000	row 01
pixe15/pixel16	0x0000_0000	row 01
Row 01 trailer	0x9798_fbfd → **Changed from right figure as below** => 16bit (0x9798) + 16bit(“1110” (4bit) + [event_type: 2bit = “10”(fixed)] + [row count: 6bit] + [internal event counter: 4bit])	row 01
Row 02 header	0x5753_fcfe	row 02
pixel01/pixel02	0x0001_0001	row 02
pixel03/pixel04	0x0001_0001	row 02
.....	.....	.....
.....	.....	.....
pixel15/pixel16	0x002f_002f	row 48
Row 48 trailer	0x5753_fcfe	row 48
Event Trailer	0xf0f0_f0f0	



Data stream from FPGA , printed on the host PC (because of endian, “0001” => “0100”)

Point to be noted I. -- additional time stamp

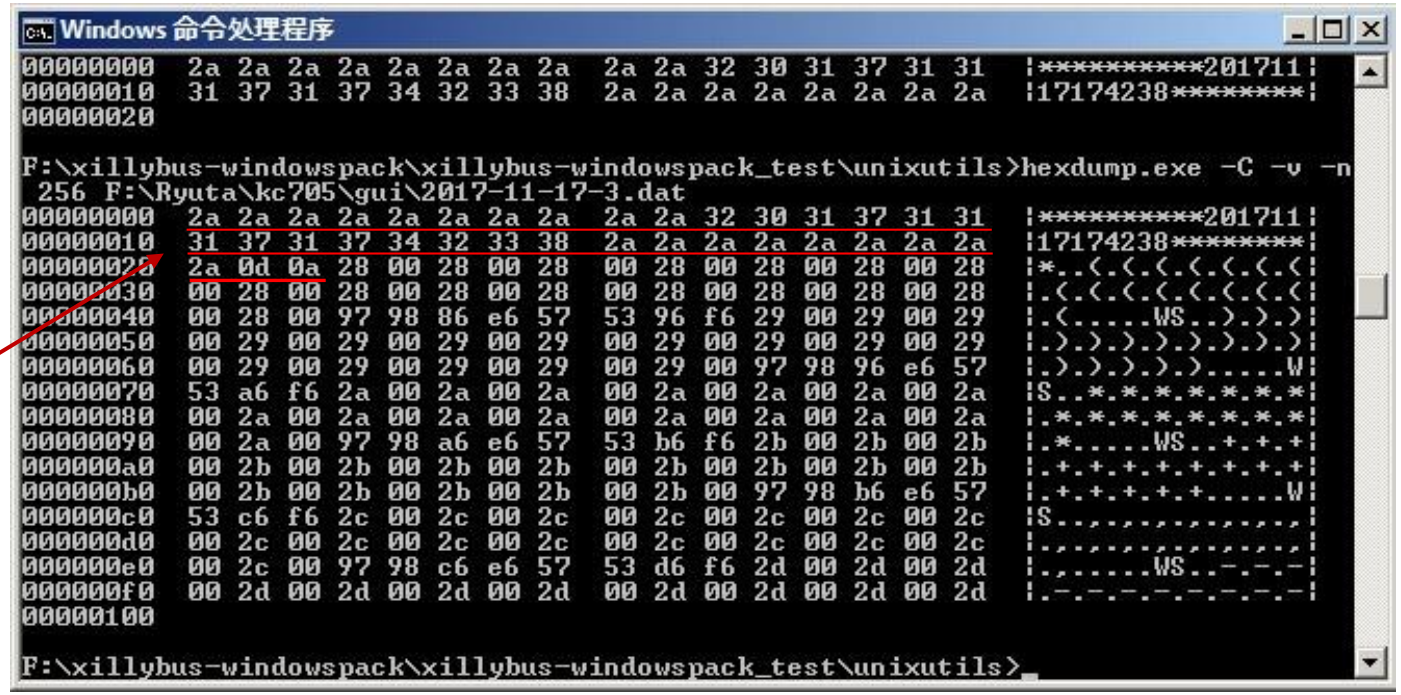
At the begging of the data, there is additional time stamp information. (prepared by Xiaoxu)

corresponding part in daq.cpp

```
177     std::string ssFilename = m_dataTaker->m_filename;
178     std::cout << "Filename = " << ssFilename.c_str() << std::endl;
179     outf = fopen(ssFilename.c_str(), "wb");
180
181     //add datetime to the beginning of the file by Lu
182     time_t t=time(0);
183     char tmp[64];
184     strftime(tmp, sizeof(tmp), "**********%Y%m%d%H%M%S**********", localtime(&t));
185     fwrite(tmp,33,1,outf);
186     fwrite("\r\n",2,1,outf);
187     while (!m_stop) {
188
```

total 35 bytes

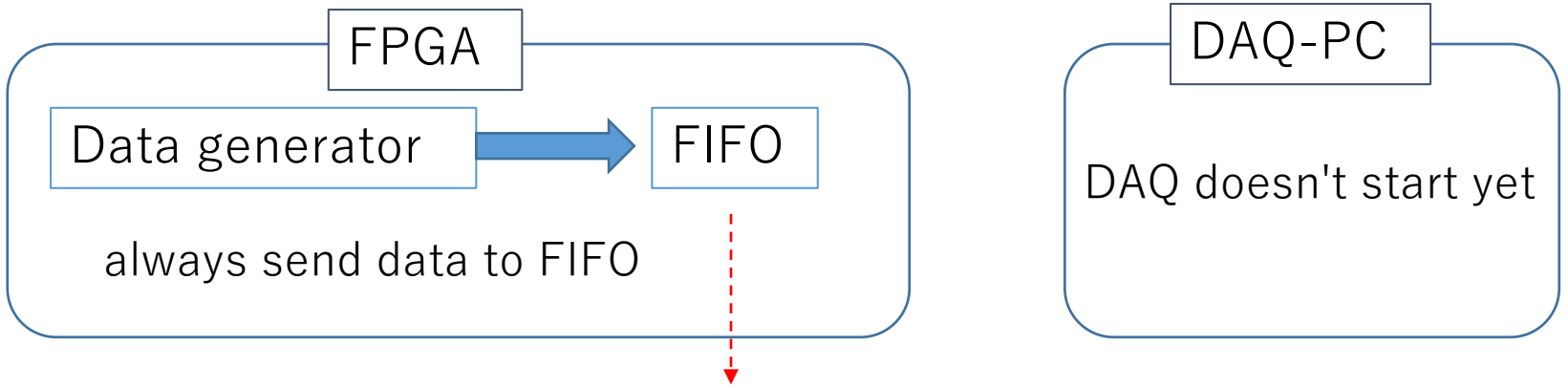
Total 35 bytes (words) are for that, and need to consider(cut) in the analysis code



Point to be noted II. -- data discontinuous point

1. Before running DAQ

FIFO : 32bit * 32768(=2^15)



FIFO becomes “full” when the DAQ is stopping/waiting and no more data can be pushed into the FIFO. However, data generated by “data generator” continuously are counting up (# this is only current(my) setting)

2. Running DAQ

Then, running DAQ and start to save the data to the harddisc.

The first 32bit(4byte)*32768 ~ 131kByte is continuous, and then, there exist a gap. If DAQ really saves every data from FPGA, then , no more data gap appears beyond this point.