A Time to Digital Converter (TDC) for g-2 experiment
Fermilab Summer Internship

Student: Ergys Hajkaj
Supervisor: Wu Jinyuan

September 2013

Fermilab

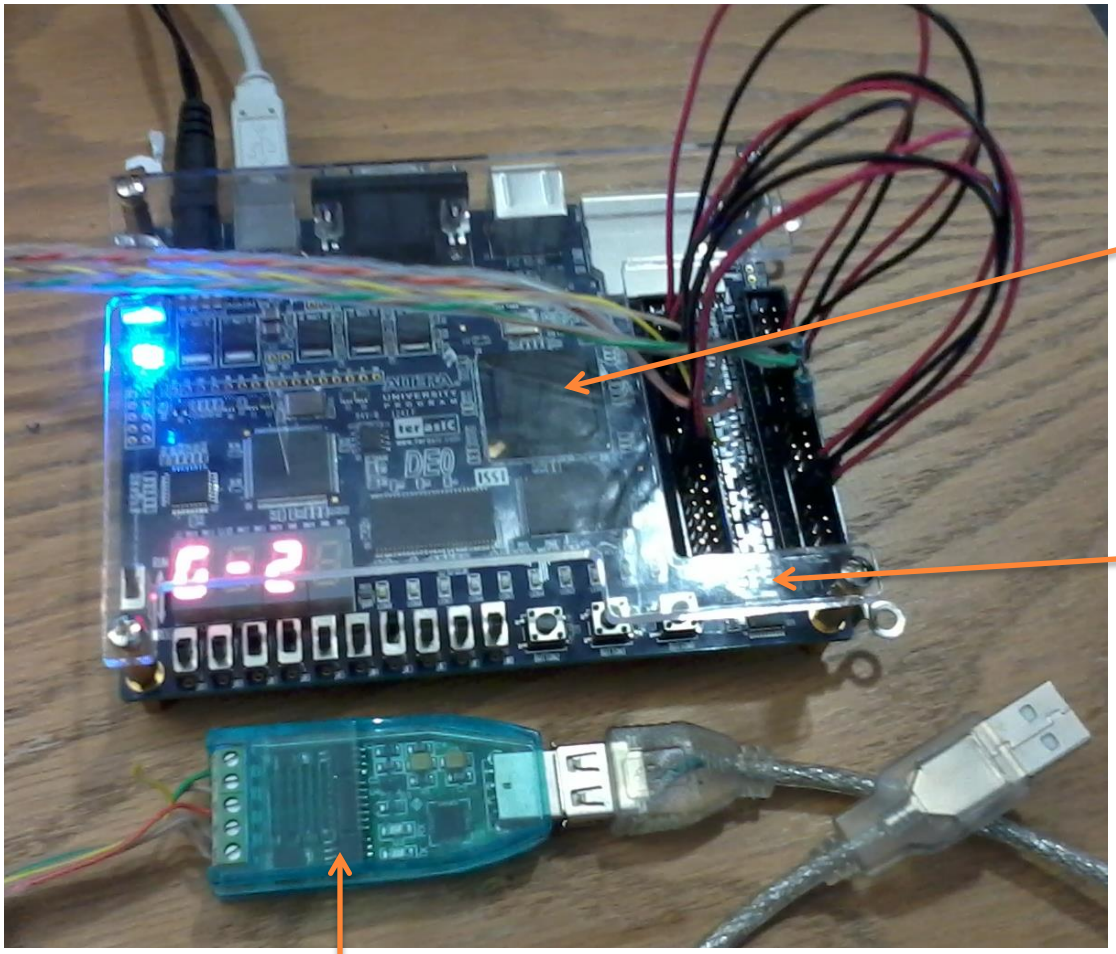
Outline

- Introduction
- First part – Study of the existing TDC firmware
- Second part – Interfacing the FPGA output data to the PC by the RS422 to USB converter
- Last part – Firmware testing using the DE0 board.
- Conclusions

Introduction

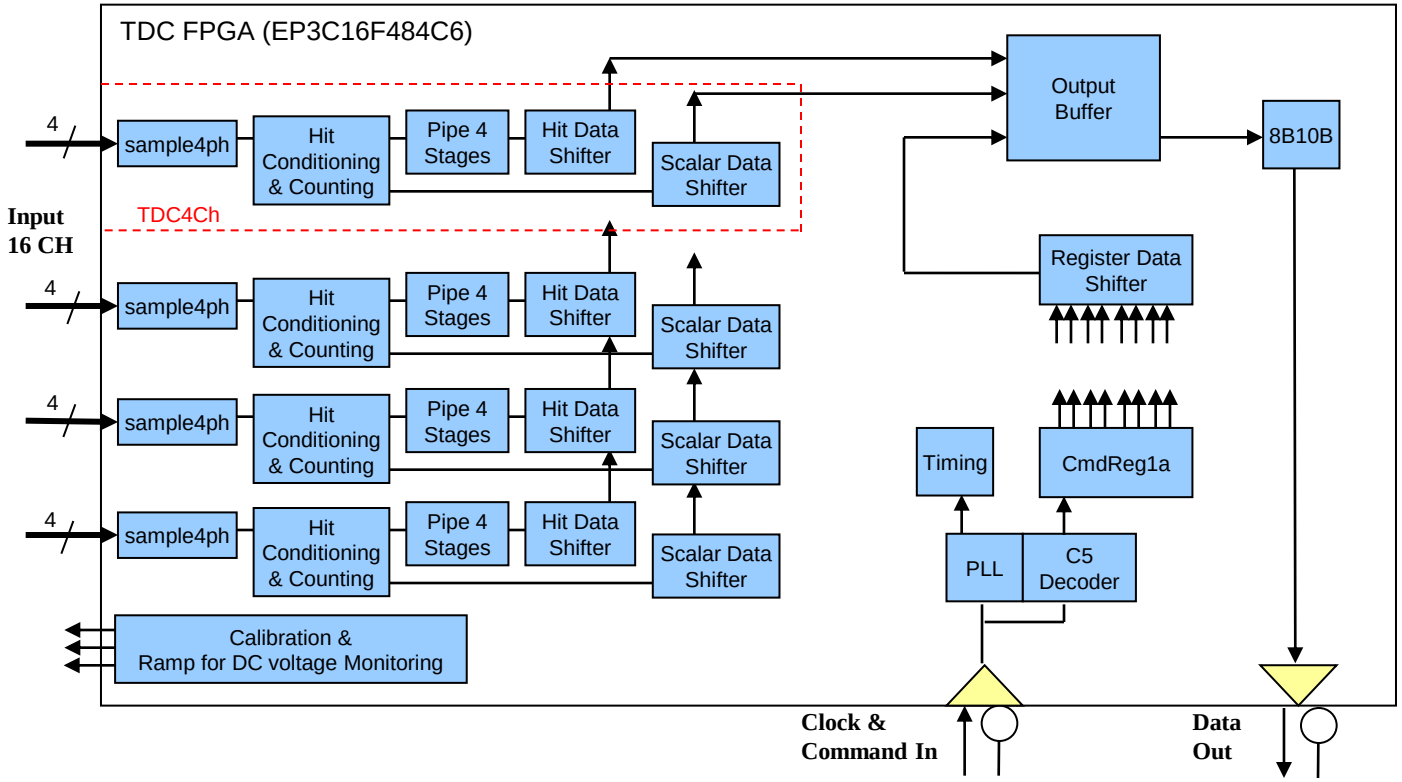
- The first part of the work was the study of the existing TDC firmware in order to debug it and make it work.
- The second part was to interface the FPGA output data to the PC using the existing RS422 to USB converter.
- The last part of the work was the firmware testing using the existing DE0 board.

The DE0 Board

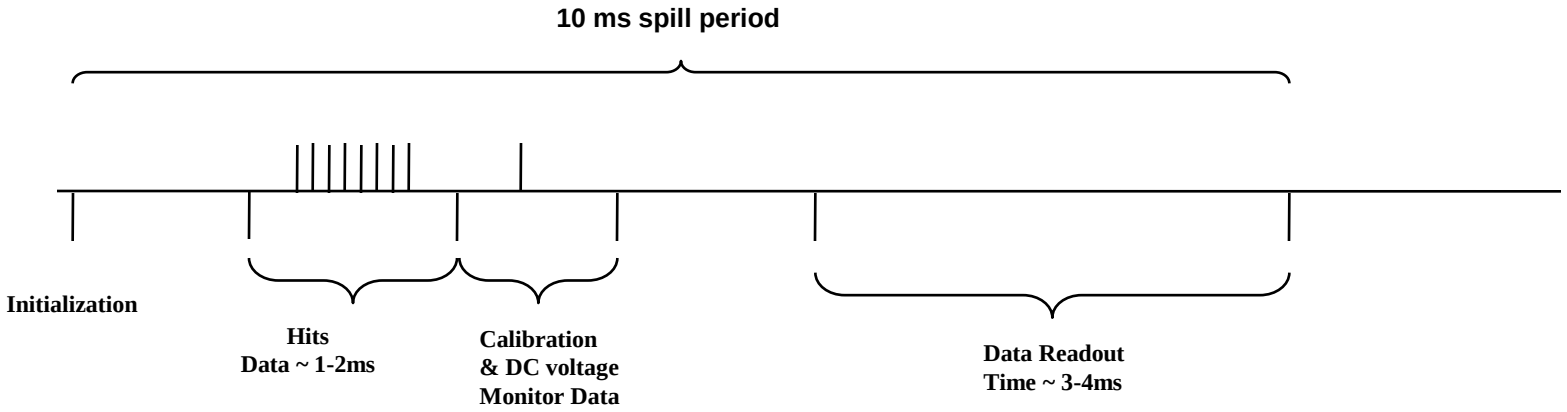


Firmware Specifications

- A 16-channel TDC fits in EP3C16F484C6 with ~50% silicon resource usage.
- Input clock at 10 MHz is multiplied to obtain 200 MHz and 400 MHz for internal operation.
- Both positive and negative input transitions are digitalized with a precision of 625 ps. (1/(4*400 MHz)).
- Each channel is equipped with a counter for events counting.
- For each 10 ms spill, up to 2016 TDC hit data, scalar data, internal register data and an ID header are packed into 2048 32-bit words and are output using 8B10B protocol at 25 Mb/s.



Normal Operation

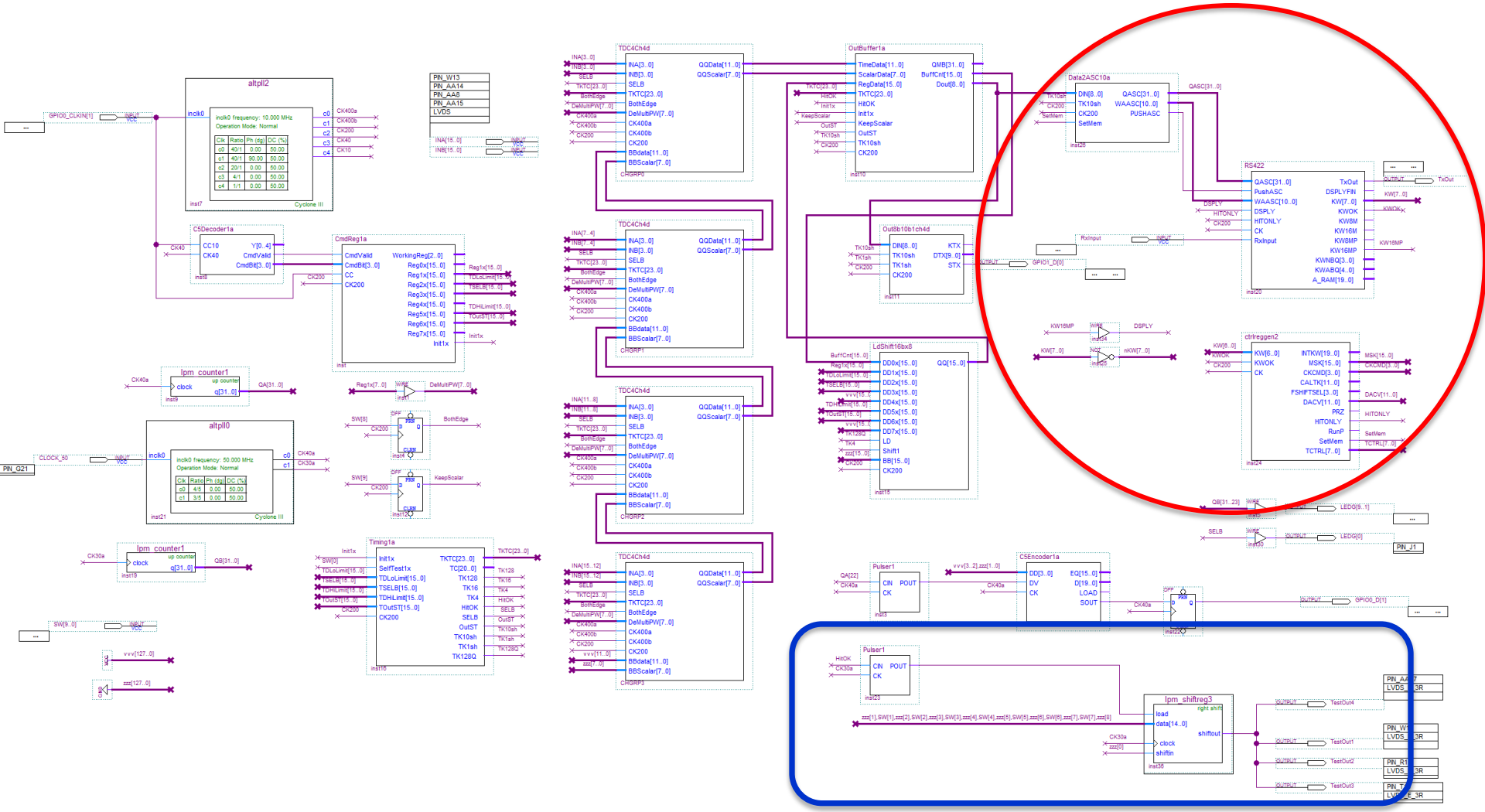


- At the start of the 10 ms spill period, a initialization command is sent to the TDC system that reset the operating sequence.
- After the initialization, the hit data are collected during the user specified time range.
- Then, the TDC values for calibration and DC voltage monitoring are collected.
- Approximately halfway of the spill period, data are sent out to the readout controller.
- The TDC system then stays in the idle state, until being initialized again for next spill.

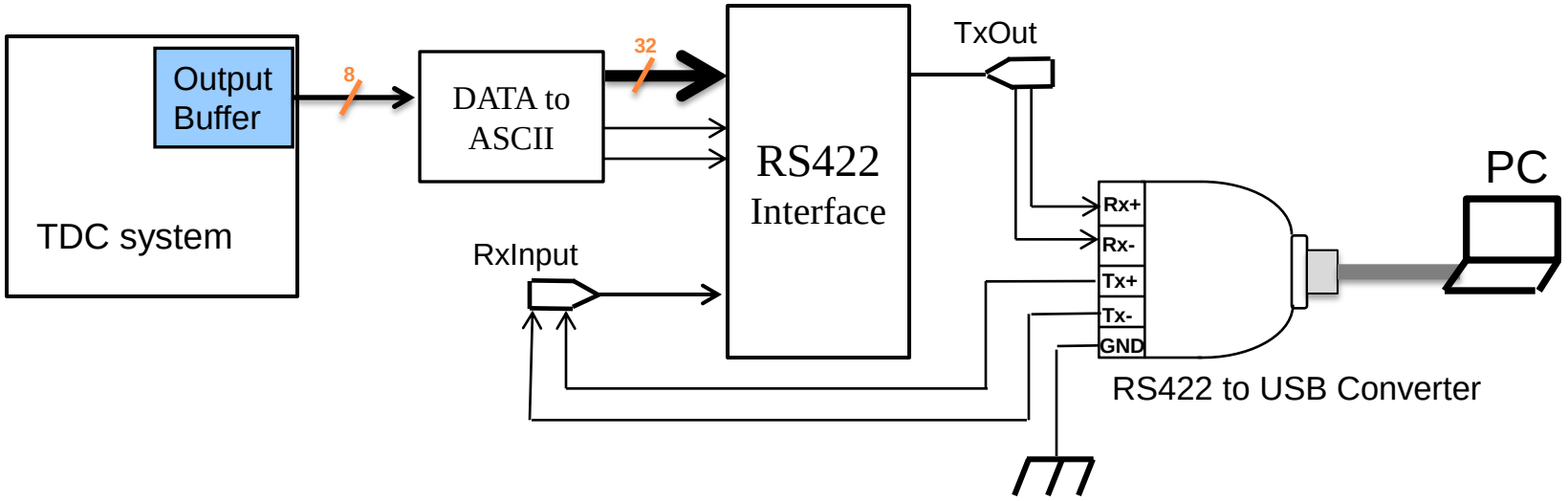
Output Data Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0																														0x3C, K28.1		K code & ID block 0-7
0																														0x67 = ‘g’		
0																														0x2D = ‘-’		
0																														0x32 = ‘2’		
0																														0x54 = ‘T’		
0																														0x44 = ‘D’		
0																														0x43 = ‘C’		
Version date, e.g., 0x20130401																																
0																								Register 7								Internal Register 8-15
0																								Register 6								
0																								Register 5								
0																								Register 4								
0																								Register 3								
0																								Register 2								
0																								Register 1								
0																								Register 0 = Number of non-empty 32-bit words								
Scalar Count 15																																Scalar Readout 16-31
Scalar Count 14																																
Scalar Count 0																																
0	0	1	Edge	CH: 0-15																												Hits 32-2047
Coarse Time: LSB= 5 ns, Range = 0 - 10.485775 ms																														Fine Time LSB=625ps		
0	0	1	Edge	CH: 0-15																												
Coarse Time: LSB= 5 ns, Range = 0 - 10.485775 ms																														Fine Time LSB=625ps		

Actual Top Page



Interfacing the FPGA output data to the PC(1)



- To view the output data of the TDC system on the PC, the data must first be converted to ASCII code.
- This function is performed by the "*DATA to ASCII*" block.
- This block also generates in output the memory addresses and the write enable signal for the next block, where the converted data is stored.
- Data are sent out the "*RS422 Interface*" block in parallel at 2.5 MB/s.

Interfacing the FPGA output data to the PC(2)

- Once converted the data into ASCII code, the "*RS422 Interface*" block is necessary so that the data is displayed correctly on the PC.
- This block is necessary because the data are output via the RS422 serial protocol.
- So this block:

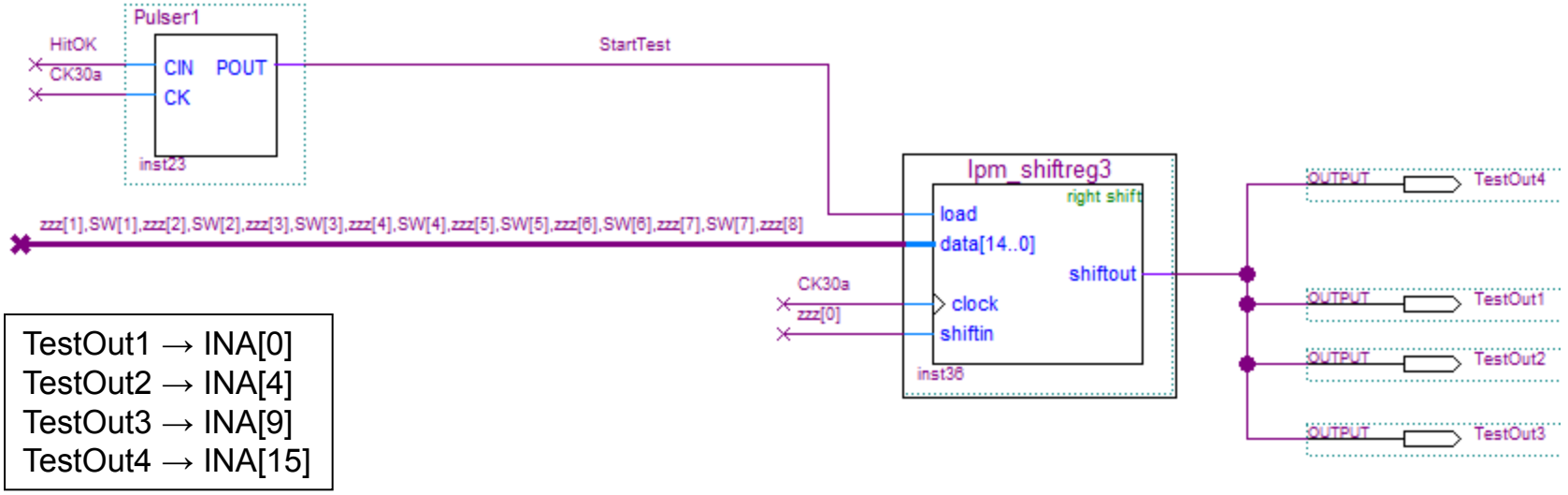
  - receives an input a data stream at a speed of 2.5 MB/s

  - stores the input data stream

  - provides as output a data stream at a speed of 0.115MB/s (that is the Baud Rate max)

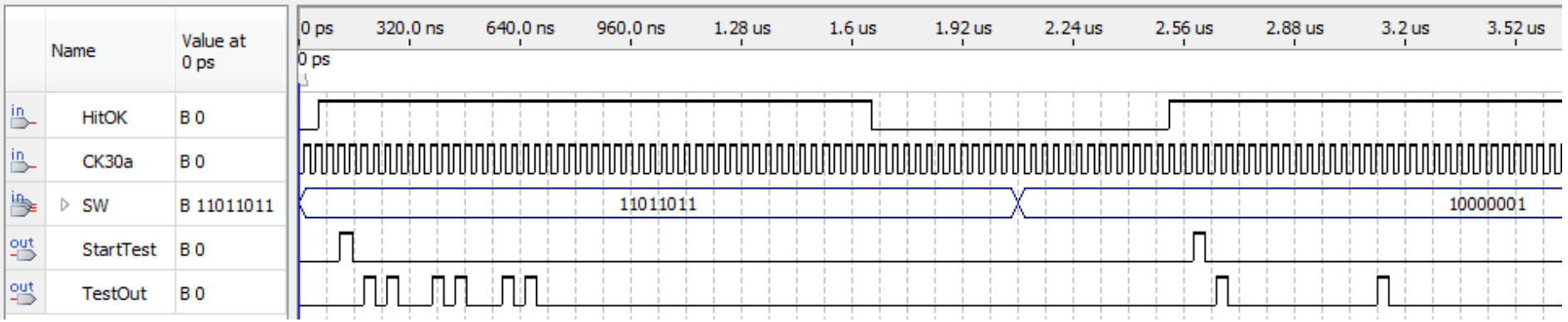
- Then, using the converter RS422 to USB, the data is sent via USB to the PC, and through the "Tera Term" program display the data on the monitor.
- Pressing the w character of the keyboard, a set memory command is send and it lets the data of a new spill period be displayed.

Testing TDC firmware (1)

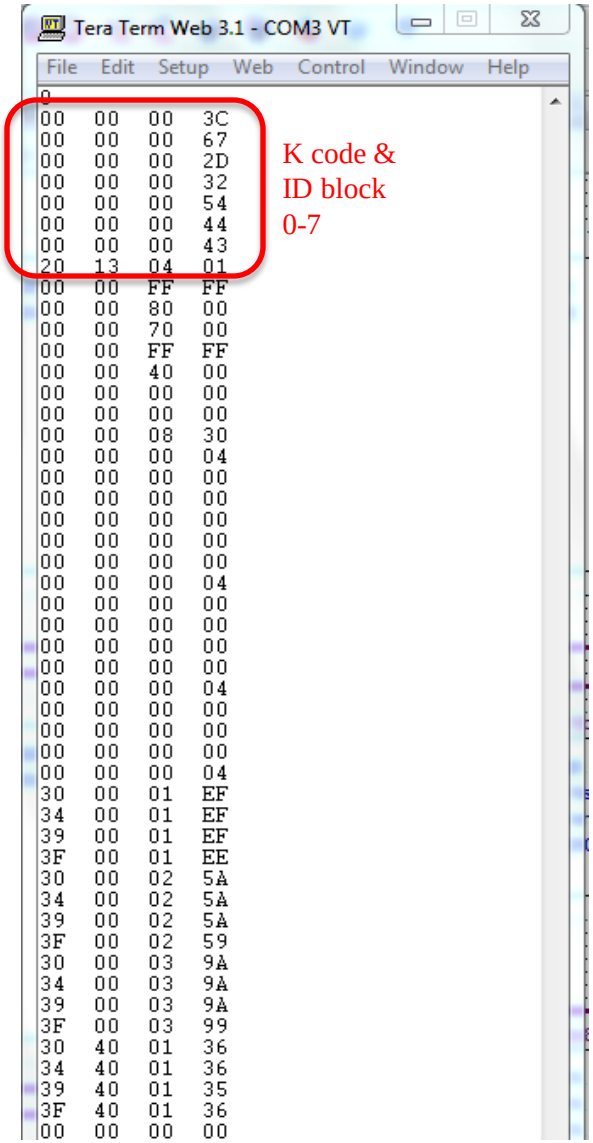


TestOut1 → INA[0]
TestOut2 → INA[4]
TestOut3 → INA[9]
TestOut4 → INA[15]

- The period of two consecutive hit is 66.66ns, this allows me to check if the time measured by the TDC is correct or not.

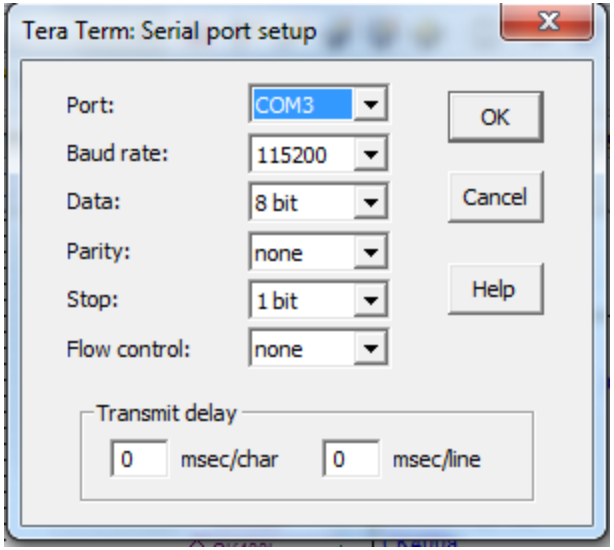


Testing TDC firmware (2)

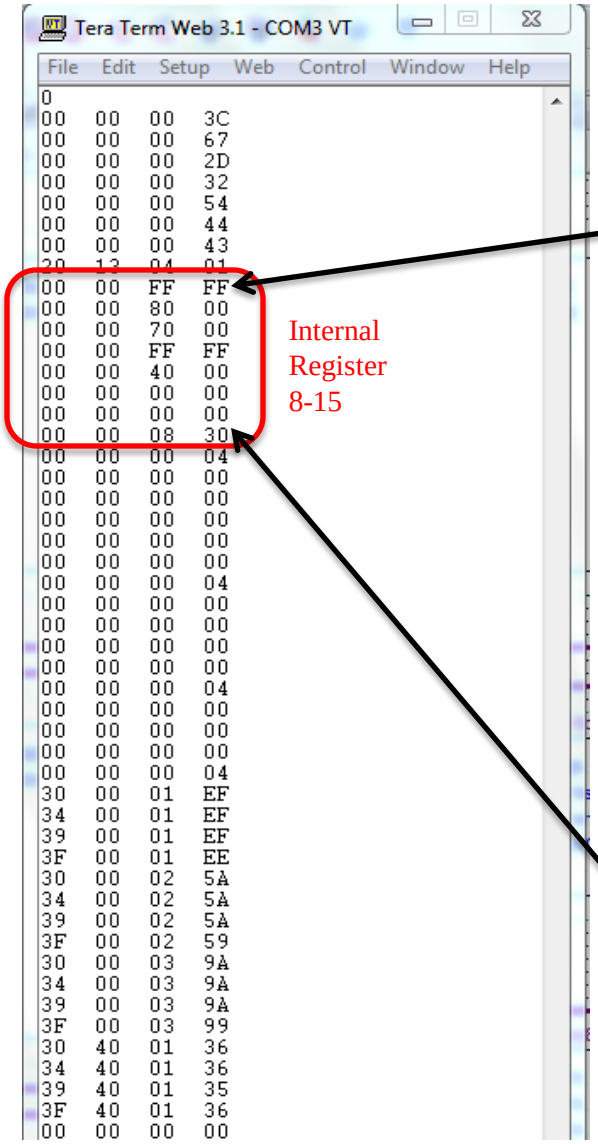


K code & ID block 0-7

7	6	5	4	3	2	1	0
0x3C, K28.1							
0x67 = 'g'							
0x2D = '-'							
0x32 = '2'							
0x54 = 'T'							
0x44 = 'D'							
0x43 = 'C'							
Version date, e.g., 0x20130401							

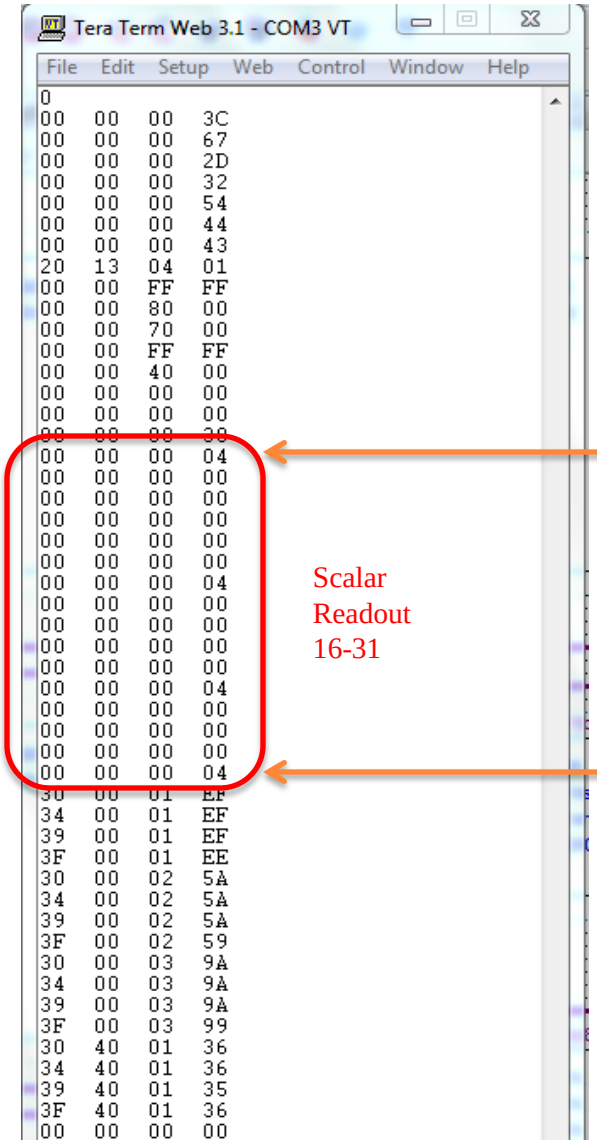


Testing TDC firmware (3)



Register	Bit(s)	Function
Reg7x		Not used
Reg6x	15..0	TOutST[15..0]
Reg5x	15..0	TDHiLimit[15..0]
Reg4x		Not used
Reg3x	15..0	TSELB[15..0]
Reg2x	15..0	TDLoLimit[15..0]
Reg1x	15..0	DeMultiPW; Pulse width for de-multi-hit function, 0-255 x 5 ns
	8	BothEdge
	12	KeepScaler
Reg0x	15..0	Number of non-empty words

Testing TDC firmware (4)



Scalar Readout 16-31

Scalar Count 15

.

.

.

.

Scalar Count 0

TestOut1 → INA[0]
TestOut2 → INA[4]
TestOut3 → INA[9]
TestOut4 → INA[15]

Testing TDC firmware (5)

Tera Term Web 3.1 - COM3 VT

File Edit Setup Web Control Window Help

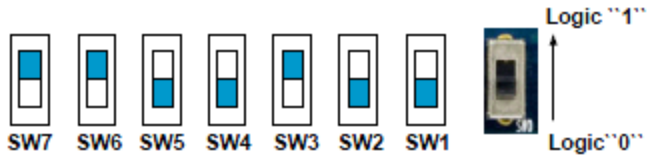
```
0
00  00  00  3C
00  00  00  67
00  00  00  2D
00  00  00  32
00  00  00  54
00  00  00  44
00  00  00  43
20  13  04  01
00  00  FF  FF
00  00  80  00
00  00  70  00
00  00  FF  FF
00  00  40  00
00  00  00  00
00  00  00  00
00  00  08  30
00  00  00  04
00  00  00  00
00  00  00  00
00  00  00  00
00  00  00  00
00  00  00  00
00  00  00  04
00  00  00  00
00  00  00  00
00  00  00  00
00  00  00  00
00  00  00  04
00  00  00  00
00  00  00  00
00  00  00  00
30  00  01  EF
34  00  01  EF
39  00  01  EF
3F  00  01  EE
30  00  02  5A
34  00  02  5A
39  00  02  5A
3F  00  02  59
30  00  03  9A
34  00  03  9A
39  00  03  9A
3F  00  03  99
30  40  01  36
34  40  01  36
39  40  01  35
3F  40  01  36
00  00  00  00
```

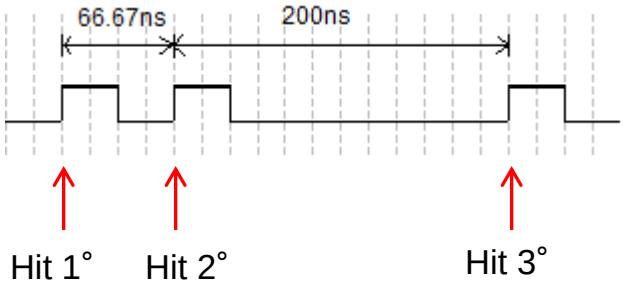
Hit 1°

Hit 2°

Hit 3°

Hits 32-47





$$25A - 1EF = 6B_{16} = 107$$
$$107 * 0.625ns = 66.87ns$$

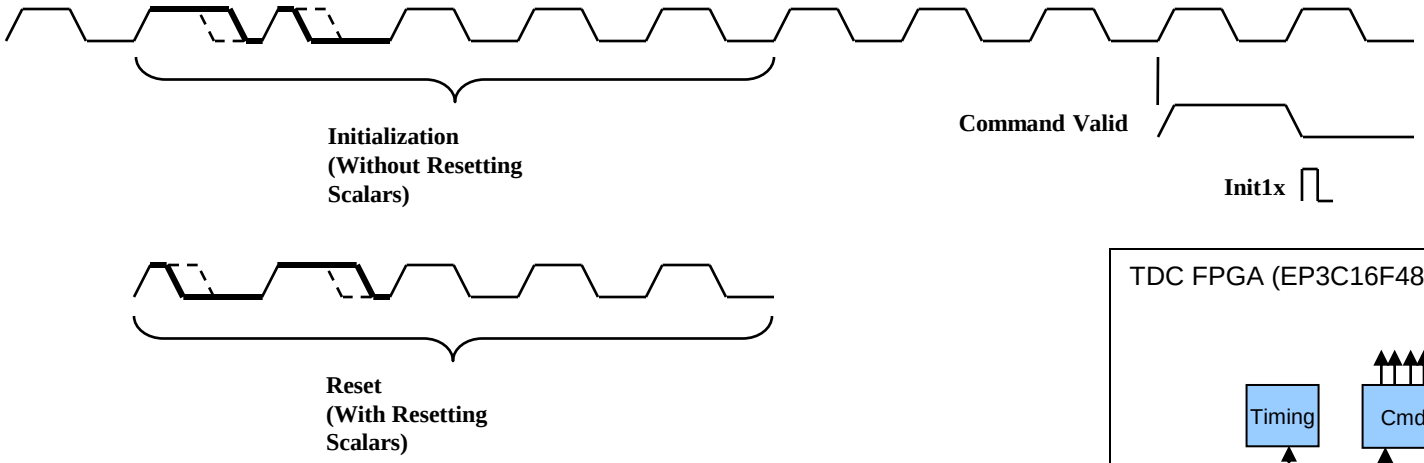
$$39A - 25A = 140_{16} = 320$$
$$320 * 0.625ns = 200ns$$

Conclusions

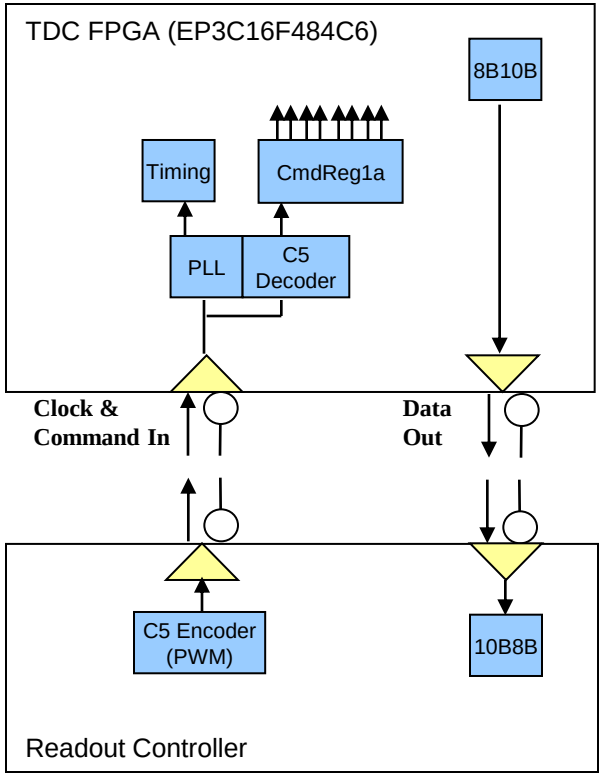
- The results shown before were possible only after correcting some aspects of the original firmware.
- In fact, testing the firmware, I found some bugs in the block *Output Buffer* that did not allow the proper functioning of the system, that I solved together with my supervisor.
- Although there are still some small things to fix, we can be satisfied of the operation of the TDC, since the main function (the measure of the time between two or more events) is carried out correctly.


Thanks to Fermilab and INFN of Pisa for allowing me to make this Internship and my supervisor Wu Jinyuan who followed and helped me in this work.
Fermilab

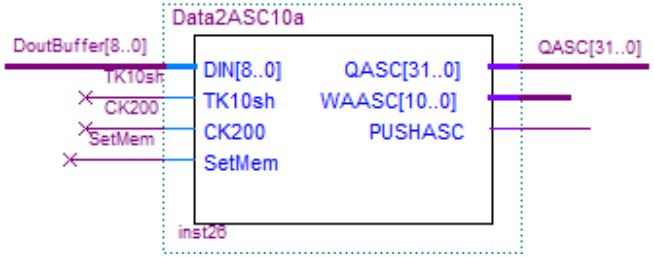
Initialization of Spill



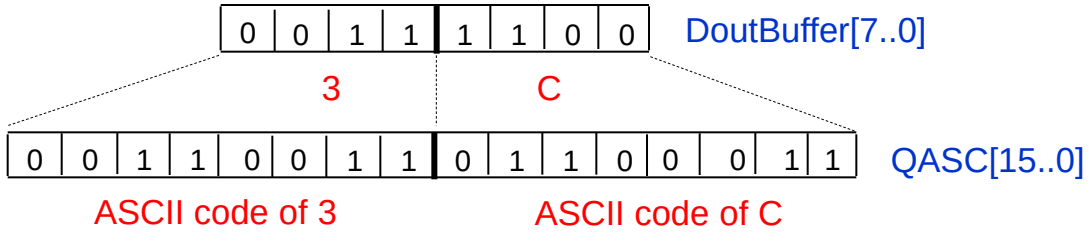
- A wide-narrow sequence is decoded as a initialization command without resetting scalars.
- The narrow-wide sequence can be reserved as resetting command that will reset scalars.
- After a known latency, an initialization pulse is generated inside FPGA that resets the coarse time counter and a normal operation sequence is started.



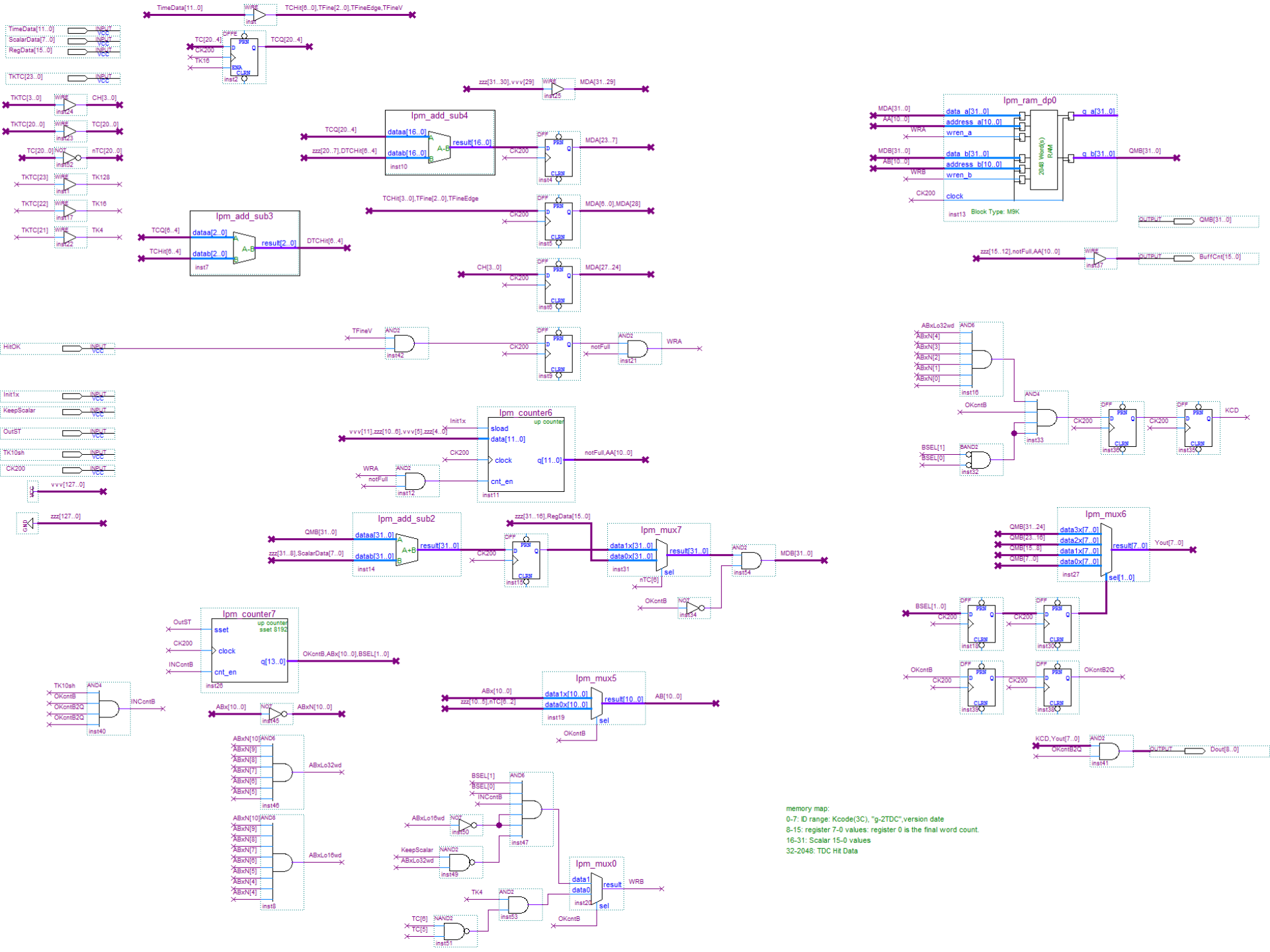
Converting data into ASCII

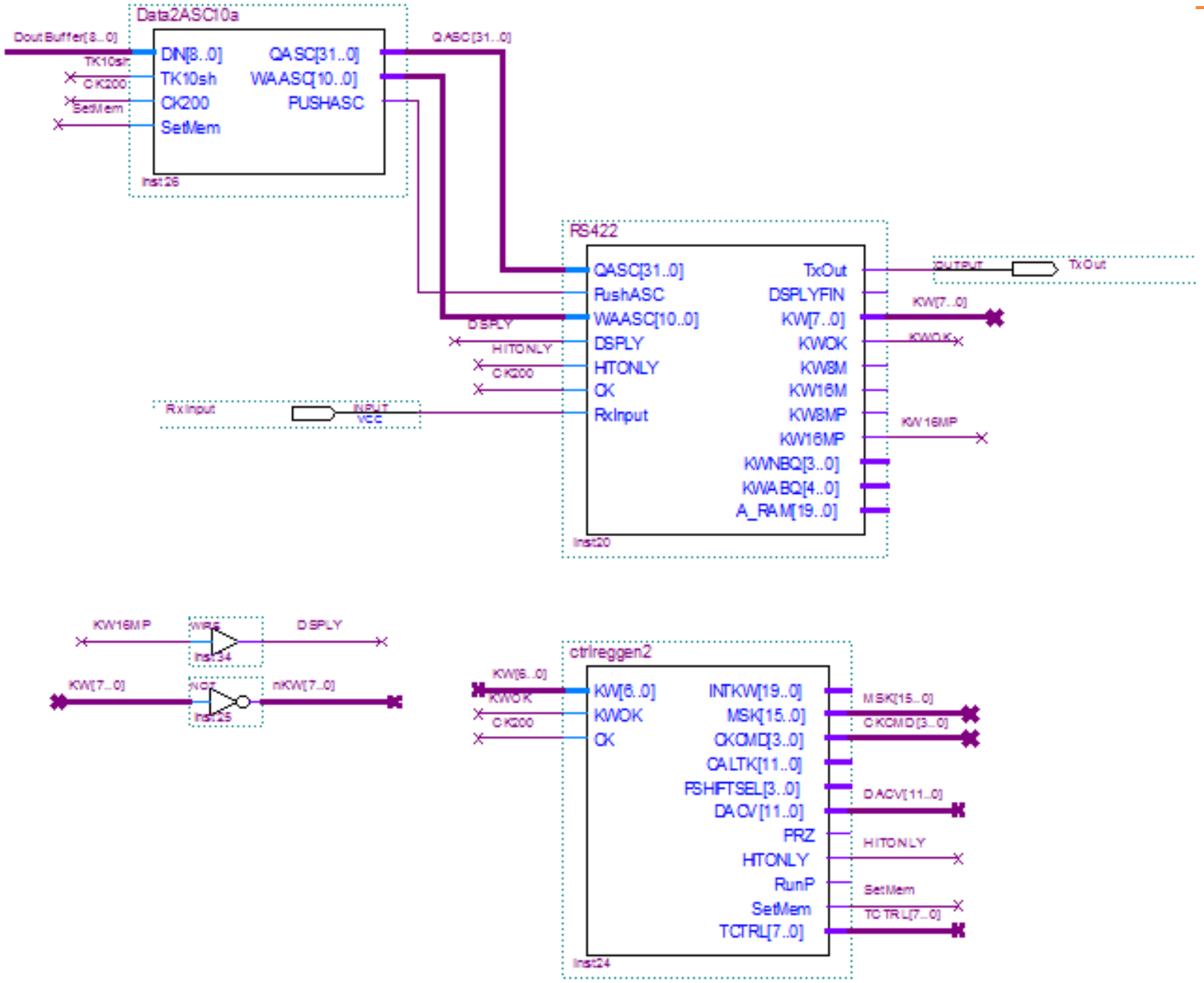


- The Data2ASC10a block converts in ASCII code the output bytes coming from the block *OutBuffer1a*.



- Each output sequence starts with a K28.1 code (DoutBuffer[8] = 1) so that the receiver can be aligned, I can use this bit to synchronize it on the first useful data to be converted.
- QASC[31..16] contains the ASCII code of two characters "SP" (space).
- WAASC[10..0] is the memory address where the output data is written.
- PushASC is the signal that enables the writing of the data in the memory.
- Data are sent out in parallel at 2.5 MB/s.


lpm_add_sub4
lpm_add_sub3
lpm_ram_dp0
lpm_counter6
lpm_counter7
lpm_add_sub2
lpm_mux7
lpm_mux6
lpm_mux5
lpm_mux0
memory map:
0-7: ID range: Kcode(3C), "g-2TDC",version date
8-15: register 7-0 values: register 0 is the final word count.
16-31: Scalar 15-0 values
32-2048: TDC Hit Data


Data2ASCI0a
DoutBuffer[8..0]
TK10sh
CK200
SetMem
DIN[8..0]
TK10sh
CK200
SetMem
QASC[31..0]
WAASC[10..0]
PUSHASC
Inst26
QASC[31..0]
RS422
QASC[31..0]
PushASC
WAASC[10..0]
DSPLY
HITONLY
CK200
RxInput
INPUT
VCC
RxInput
TxOut
DSPLYFIN
KW[7..0]
KWOK
KW8M
KW16M
KW8MP
KW16MP
KWNBQ[3..0]
KWABQ[4..0]
A_RAM[19..0]
Inst20
TxOut
KW[7..0]
KWOK
KW16MP
KW16MP
WIRE
DSPLY
Inst34
KW[7..0]
NOT
nKW[7..0]
Inst25
ctrlreggen2
KW[8..0]
KWOK
CK200
KW[8..0]
KWOK
OK
INTKW[19..0]
MSK[15..0]
CKCMD[3..0]
CALTK[11..0]
PSHIFTSEL[3..0]
DACV[11..0]
PRZ
HITONLY
RunP
SetMem
TCTRL[7..0]
MSK[15..0]
CKCMD[3..0]
DACV[11..0]
HITONLY
SetMem
TCTRL[7..0]
Inst24