

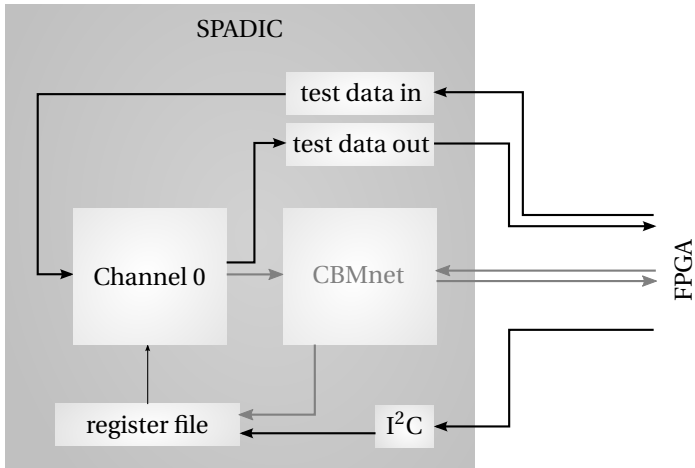
SPADIC 1.0 – data path tests

Michael Krieger

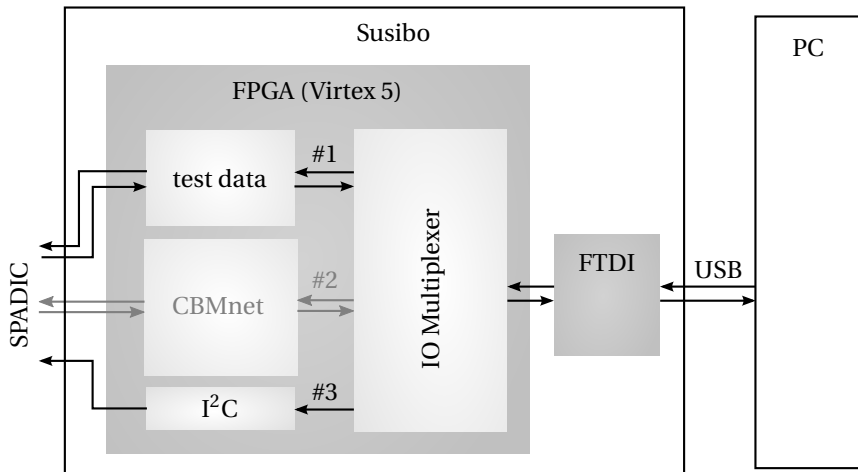
CBM DAQ Meeting, 09.05.2012

- Test setup overview
- Tests of specific features
 - Test data input
 - Hit detector
 - Hit window & selection mask
 - Filter settings
- Summary

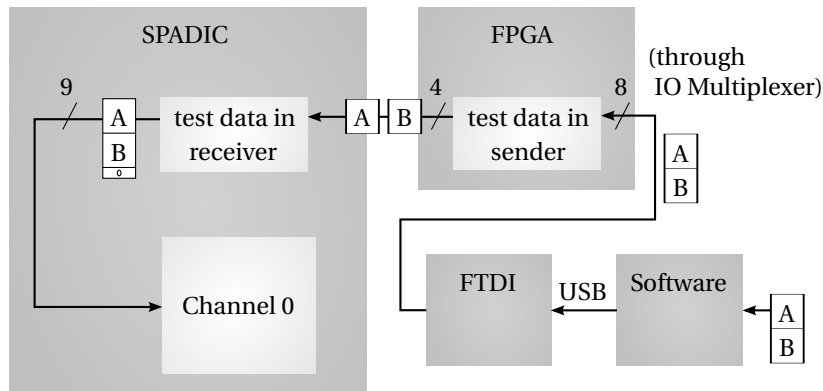
Test setup (SPADIC part)



Test setup (FPGA part)

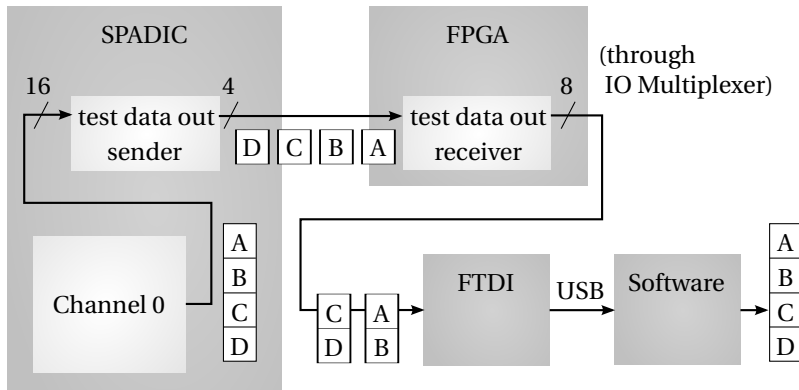


Test data input protocol



- 0xAB is sent
- $(0xAB \ll 1)$ received at channel input (replaces ADC values)
→ can only send even numbers

Test data output protocol



- FTDI handles 8 bit data
→ final reassembly of 16 bit message words done in software

Does the test data input work?

- already knew that data *output* works from Tim's analog tests

procedure

- send some arbitrary values to data input and look for any data coming out of the FTDI
- first result:
 - data from FTDI received in response to the input ✓
 - but: no valid messages found

Analysis of received data

example data from FTDI

A0 00
BE E3
A2 20
20 00
22 A9
63 26
2C 44
B2 33
A0 00
BE EB
AE 36
25 22
23 3D

this would make sense

80 00 ← start of message
9E E3 ← timestamp
A2 20 ← start of data
20 00
22 A9
63 26
2C 44
B2 33 ← end of message
80 00 ← start of message
9E EB ← timestamp
AE 36 ← start of data
25 22
23 3D

suspicion

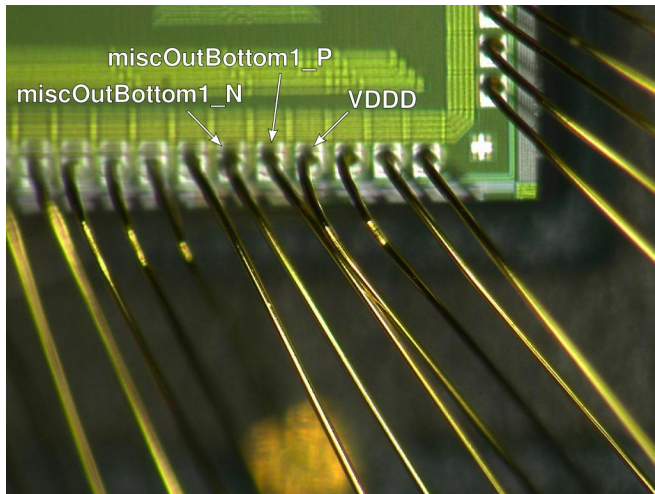
wrong hex-digits:

- they are
2, 3, 6, 7, A, B, E, F
- when they should be
0, 1, 4, 5, 8, 9, C, D

explained if one of the four bits
is always 1:

..1.

Solution: touching bond wires



Now it works

After fixing a few other things in the firmware ...
(timing/misalignment of 4 bit ↔ 8 bit)

Python software:

- same basic functionality as *Hitclient*
- interactive session → particularly good for debugging and testing

```
>>> s = Spadic()
>>> s.write_data([100, 150])
>>> s.write_data([100, 150])
>>> s.write_data([100, 150])
>>> s.write_data([100, 150])
>>> M = [Message(m) for m in messages(message_words(s.read_data()))]
>>> len(M)
4
>>> print M[-1].data
[0, 0, 100, 150, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Testing the hit detector

- two thresholds
- normal and differential mode → they work ✓

bug in the digital comparator logic:

```
>>> s.threshold(-20, -10)
>>> s.write_data([100, 150])
>>> s.write_data([100, 150])
>>> s.write_data([100, 150])
>>> s.write_data([100, 150])
>>> M = [Message(m) for m in messages(message_words(s.read_data()))]
>>> len(M)
0
```

- $\text{threshold} < 0 < \text{value}$ → no hit is detected
- all other cases are correct
- → cannot “disable” one of the thresholds by setting it to -255, must always use both thresholds

Testing the message builder

Choose how many & which values to include

- hit window length
- selection mask

Works in principle, but:

for certain combinations of hit window/mask

wrong and/or missing values

Example of wrong and missing values

- hit window length = 32
- selection mask = all selected
- offset = 205
- input values = [20, 10]

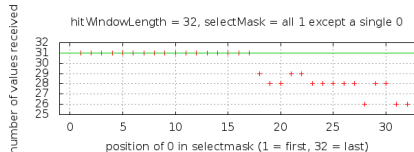
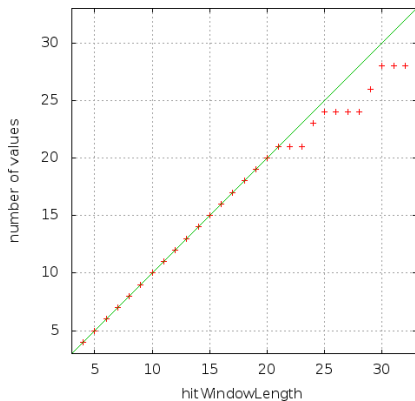
expected: 32 values in the message

[205, 205, 225, 215, 205, ..., 205]

result: 28 values (but 32 reported)

[205, 205, 225, 215, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, 205, -167, -167, -167, -149, 107, 107, 109, 205, 205, 0]

Number of values in a message for different settings



Explanation for wrong and missing values

test data output sender (on SPADIC)

- situation where input buffer is “almost empty” is not handled correctly
- message words are lost
- window length & mask settings determine “rhythm” of message words fed into the buffer

Message building works properly

→ all should be fine if CBMnet is used!

Setting the filter coefficients

Test

- set the same coefficients for all 4 stages
- enable one stage at a time
- send some values in, e. g. [100, 110]
- outcome should be the same

Result

expected: [0, 100, 131, -29, 0]

- [0, 100, 91, -21, 0]
- [0, 100, 88, -25, 0]
- [0, 100, 88, -25, 0]
- [0, 100, 88, -25, 0]

Setting the filter coefficients *right*

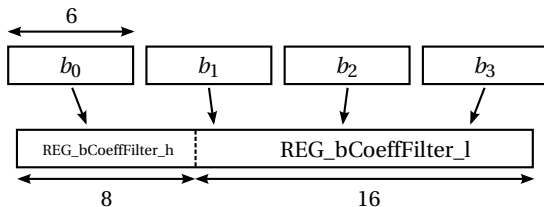
wrong:

```
sum(c << 6*i for (i, c) in enumerate(coeff))
```

correct:

```
sum(c%64 << 6*i for (i, c) in enumerate(coeff))
```

→ handle 2's complement correctly!



- Test data input ✓
- Test data output
 - bug: message words lost
 - not relevant when CBMnet works
- Hit detector ✓
 - except bug in comparator logic: neg. thresholds
- Hit window, selection mask ✓
- Filter settings ✓