ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

ADXL371/2/3 FIFO issue and workaround

Pablo del Corro

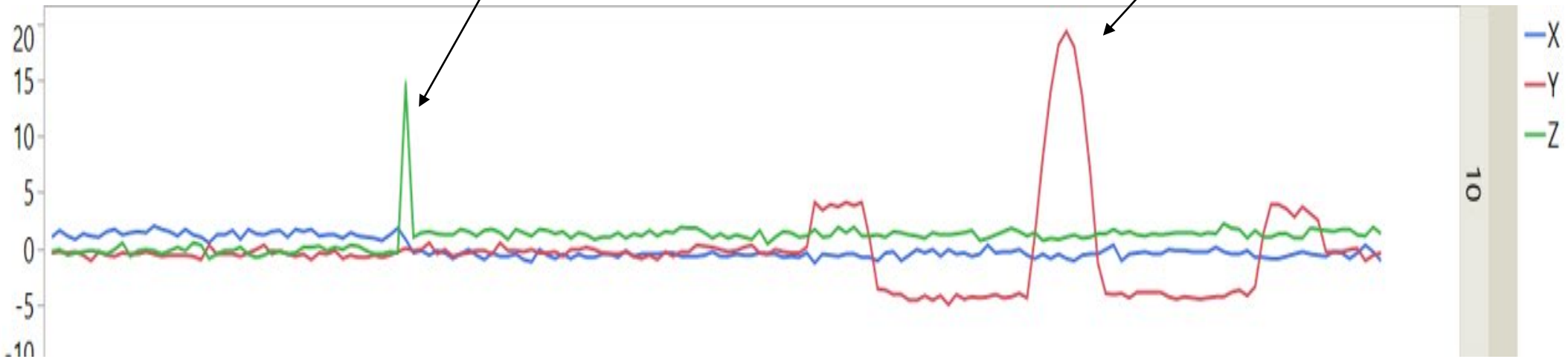
FIFO issue description

- Customer reports issue when using the FIFO samples get misaligned.
  - Expected: Data should be stored in the FIFO as X,Y,Z,X,Y,Z,… sequence.
  - Issue: In all FIFO modes data misalignment occurs. Data could be stored in the FIFO as Y,Z,X,Y,Z,X,… sequence or Z,X,Y,Z,X,Y,…
- Root cause:
  - The Asynchronous FIFO design has an arbitrator that has a functional issue.
- Details:
  - The system clock is responsible for filling the FIFO (write pointers)
  - The serial port clock is responsible for reading the FIFO (read pointers)
  - When the user reads the FIFO, the write pointer location is synchronized to the serial port
  - The system clock monitors how many entries the user reads (read pointers)
- Data misalignment happens more often when in FIFO Stream mode due to read and write pointers collision.

Example of data misalignment

Impact was applied on Z-axis, but FIFO stored data as it if happened on Y-axis

Moment when data misaligned



Example of data misalignment

- Data should be stored in the FIFO as X,Y,Z,X,Y,Z,... sequence.
- X-axis data is recognized by series start indicator (SSI) on X_low byte. Y and Z axes do not have indicator bit.
- The value on X_low is always an odd number.
- The figure on the right shows that data stored in FIFO eventually gets misplaced (highlighted in blue).
- This type of failure occurs more often in Stream Mode and does not show a pattern that can be exploit to workaround the issue.
- In Trigger/Oldest Saved Mode, the misalignment occurrence rate seems to decrease considerably.

▯ :series start indicator (odd number)

# of data	X_h	X_l	Y_h	Y_l	Z_h	Z_l	X(g)	Y(g)	Z(g)
1	0	17	255	192	250	112	0.098	-0.391	-8.691
2	255	129	0	0	250	32	-0.781	0.000	-9.179
3	0	129	255	160	251	224	0.781	-0.586	-6.445
4	255	177	0	0	254	48	-0.488	0.000	-2.832
5	0	1	0	32	0	192	0.000	0.195	1.172
6	0	177	0	0	3	176	1.074	0.000	5.761
7	255	177	0	96	5	48	-0.488	0.586	8.105
8	255	177	0	240	5	112	-0.488	1.465	8.496
9	255	49	0	144	5	64	-1.269	0.879	8.203
10	0	49	255	240	3	96	0.293	-0.098	5.273
11	0	145	255	240	0	96	0.879	-0.098	0.586
12	255	177	255	240	253	208	-0.488	-0.098	-3.418
13	255	241	255	208	251	144	-0.098	-0.293	-6.933
14	0	81	255	208	250	48	0.488	-0.293	-9.081
15	0	129	255	128	250	160	0.781	-0.781	-8.398
16	254	112	0	65	0	0	-2.441	0.391	0.000
17	2	224	255	241	0	64	4.492	-0.098	0.391
18	4	192	255	65	0	144	7.421	-1.172	0.879
19	5	224	1	33	255	240	9.179	1.758	-0.098
20	5	64	0	17	255	208	8.203	0.098	-0.293
21	3	160	255	177	0	80	5.664	-0.488	0.488
22	1	64	0	17	255	160	1.953	0.098	-0.586
23	254	96	0	17	255	160	-2.539	0.098	-0.586
24	252	0	255	241	255	128	-6.250	-0.098	-0.781
25	250	48	0	241	255	128	-9.081	1.465	-0.781
26	250	16	255	177	255	112	-9.277	-0.488	-0.879
27	251	192	0	81	255	144	-6.640	0.488	-0.684
28	253	112	0	65	0	0	-4.004	0.391	0.000
29	0	208	0	193	0	0	1.269	1.172	0.000
30	3	32	255	225	0	128	4.883	-0.195	0.781
30	4	208	0	65	255	192	7.519	0.391	-0.391

Z axis? X axis? Y axis?

Proposed Workaround

- The ADXL372 allows the use of external clock and/or external trigger signal.
  - External clock can be used for improved clock frequency accuracy or to control ODR (tied to INT1)
  - External trigger can be used for synchronize acceleration sampling to an external signal (tied to INT2)

- Proposal: Leverage the external trigger synchronization function to disable the sensor ADC before accessing the FIFO.
  - Force INT2 to LOW, always stays LOW.
  - Configure XL37x in internal trigger mode and the desired FIFO mode. ODR is generated internally.
  - Before any FIFO read, configure the XL to external trigger mode. Since no trigger provided (INT2 = LOW), ADC is now inactive, and no FIFO write will take place
  - Commence FIFO read
  - After FIFO read completes, configure the XL to internal trigger mode. Sensor starts sampling data again.
- Pros and Cons of workaround:
  - Pros: eliminates the FIFO issue for all FIFO modes, no restrictions on ODRs or need of data manipulation to "re-align" the data by using the series start indicator (SSI) bit
  - Cons:
    - INT2 = GND. Cannot be used to map any interrupt.
    - The time the ADC is OFF, is a "blind time" where no data is being acquired.

Workaround implementation example

- **Initialization:**
  - Set desire FIFO mode and other desired configurations.
  - Set the timing register (0x3D) to external sync along with the ODR
  - **No external trigger signal is applied to INT2 to keep the ADC off**
  - Set to Measurement mode.
- **Loop N times:**
  1. Set the timing register (0x3D) to internal sync along with the ODR
  2. Wait for FIFO_FULL interrupt from INT1
  3. Set the timing register (0x3D) to external sync along with the ODR
  4. Read entire content of FIFO
  5. Clear FIFO (Bypass mode)
  6. Back to desire FIFO mode