# Progress on RD53B (ITkPix-v1) testing 

Daniel Antrim, Maurice Garcia-Sciveres, Timon Heim, Hongtao Yang

Instrumentation Weekly Meeting
Aug 7, 2020

## Introduction

- HitOR path is a basic block of RD53B digital circuit
- Self-trigger relies on inputs from hitOR and has important use cases
- Precision ToT also relies on hitOR inputs. Particularly interesting now given the known caveat in ToT memory (Maurice's talk)
- Important to verify these features on RD53B chip!



## Setup

- RD53B chip (0x10147) mounted on single chip card
- LDO powering, 1.6V for both analog and digital
- Readout using YARR system. Relying on scan results (data stream) and scope for verification



## Clear ToT memory



- Resetting ToT memory (Timon's talk) improves quality of digital scan
- Before clearing: digital current reach limit, digital scan noisy
- After clearing: digital current 1.7A, digital scan clean
- Analog current 0.7A in both cases


## Verification of pixel hitOR (JIRA)

- Procedure: in digital scan, enable hitOR bit for pixel in row $0,2,4,8$ of column 0/1/2/3 (corresponding to 4 hitOR paths), and probe hitOR0/1/2/3 output on LVDS
- Results: things are working as expected
- See 10 spikes corresponding to 10 digital injections, pattern repeat for 4 times for 4 pixels on
- HITOR_MASK0/1/2/3, when switched to 1 , will block the hitOR signal from corresponding core column
- Timon measured relative delay of hitOR signal do be order of 2.5 ns


|  | 0 | 1 | 2 | 3 |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | (1) | (2) | (3) | 4 | (1) | (2) | © | 4 |
|  | 3. | $2 \times$ | (1) | '21 |  | 4 | (1) | (2) |
| 2 | (1) | (2) | (3) | 3 | (1) | (2) | (3) | 4 |
|  | (3) | $4 \times$ | (1) | 2 | (3) | 4 | (1) | (2) |
| 4 | (1) | (2) | (3) | 4 | (1) | (2) | ( | 4 |
|  | (3) | $4 \times$ | (1) | 2 | (3) | 4 | (1) | (2) |
| $8$ | (1) | (2) | (3) | 4 | (1) | (2) | (3) | 4 |
|  | (3) | 4 | (1) | (2) | (3) | 4 | (1) | (2) |

## Self-trigger digital scan (JIRA)

- Perform self-triggered digital scan with following configurations
- Self-trigger digital threshold 1, multiplier 16, delay 45, latency 60
- Self-trigger digital threshold verified: working as expected
- The rest is the same as standard digital scan, except that trigger words are now suppressed in command stream
- Scan working well. Tags have values reserved for self-trigger


## Self-trigger pulse in LVDS output

| Val | Selected signal | Val | Selected signal |
| :---: | :--- | :---: | :--- |
| 0 | Serial CMD input straight form differential receiver | 21 | Self Trigger trigger pulses |
| 1 | Recovered serial CMD from CDR, before any delay | 22 | CDR/PLL mon_up_cdr signal |
| 2 | Recovered serial CMD from CDR, after phase delay | 23 | CDR/PLL mon_dn_cdr signal |
| 3 | Recovered 160 MHz clock before any delay | 24 | CDR/PLL mon_up_fd_cdr signal |
| 4 | Recovered 160 MHz clock after phase delay | 25 | CDR/PLL mon_dn_fd_cdrsignal |
| 5 | BX Clock to Pixel Matrix | 26 |  |
| 6 | Clock used for Precision ToT (640 MHz) | 27 | Serial stream from DATA_IN1 diff. Rcvr. |
| 7 | Pattern bit [3] of OUTPUT_PAD_CONFIG reg.85 | 28 | Serial stream from DATA_IN2 diff. Rcvr. |
| 8 | Pattern bit [2] of OUTPUT_PAD_CONFIG reg.85 | 29 | Serial stream from DATA_IN3 diff. Rcvr. |
| 9 | Pattern bit [1] of OUTPUT_PAD_CONFIG reg.85 | 30 | unused |
| 10 | Pattern bit [0] of OUTPUT_PAD_CONFIG reg.85 | 31 | unused |
| 11 | CalEdge as produced by the CMD Decoder | 32 | unused |
| 12 | CalAux as produced by the CMD Decoder | 33 | Power On Reset out (not used internally) |
| 13 | GlobalPulse as produced by the CMD Decoder | 34 | Channel Synchronizer Lock signal |
| 14 | Trigger pulses as produced by the CMD Decoder | 35 | PLL Lock signal |
| 15 | Read_Trigger pulses from the CMD Decoder | 36 | Activity detector on CMD input (Sec. 3.2 .2 ).2) |
| 16 | HitOr [3] | 37 | Activity detector on CMD input (Sec. 3.2 .2$)$ |
| 17 | HitOr [2] | 38 | Goes low when low power mode is activated |
| 18 | HitOr [1] | 39 | GADC end of conversion signal |
| 19 | HitOr [0] |  |  |
| 20 | HitOr logic result from self trigger block |  |  |

- According to Table 27 in the RD53B manual, we should be able to see self-trigger pulse with LVDS output value 21
- In our test, however, we could not see the pulse in 21. Instead it was observed with value 14 , which is assigned to trigger pulse from CMD decoder
- Confirmed in chip code later. Issue submitted to RD53B manual as well as chip development



## Precision ToT/ToA

- Generate PToT/PToA data with following procedure (repeated for each pulse duration)
- Clear ToT memory
- Enable pixel (col, row) $=(0,0),(1,0),(2,0),(3,0)$ (one pixel for each hit bus)
- Perform digital injection with duration N
- Readout PToT/PToA data
- Some observations
- PToT/PToA data always have the format islast $=1$, isneighbour $=0$, and qrow = 196 (unphysical value used to to identify PToT data)
- PToT core column mask is working (has to be "1" to produce data). So is PToA enable bit. However, PToT enable bit CANNOT suppress PToT data


## Deciphering PToT/PToA data

- First 11 bits are PToT, followed by 5 bit PToA
- PToT should be counted by 640 MHz rather than 1.28 GHz clock as mentioned in the manual (verified in chip code)
- When pulse width $=4 \times$ N PToT data output will be incomplete ("1111" suppression)

Example of digital injection pulse width 20


| 010 | 00000 |  |  |  |  | 0000 | 0000 | 000 | 000 | 0000 | 00 | 00 | 000 |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | (0) 100 | 000 |  | 0 | 100 | PToA1 | (1111) | 0100 | 000 |  | 0 | 100 | PToA2 | (1111) | 0100 | 000 | PToT3 | 0 | 100 | 000000 |  |  |
| NS |  |  | PToT1 |  |  |  |  |  |  | PToT2 |  |  |  |  |  |  |  |  |  | PToA3 | End | of stream |
|  | 64 | 0 | 79 | 0 | 8 | 8 | 15 | 64 | 0 | 79 | 0 | 8 | 8 | 15 | 64 | 0 | 79 | 0 | 8 | 8 |  |  |

## Verilog code

```
always ff a(posedge FastClk) begin : ToT Counter_=
        if ( CntReset )
        counterVal_notmr_nocg <= 10'b0;
        else
        if ( counterVal_notmr_nocg != 11'h7ff ) //increment if not max-count, otherwise stay at max-count
        counterVal_notmr_nocg <= counterVal_notmr_nocg + 1'b1;
    end : ToT_Counter
```


## Next steps

- Implement decoding of precision ToT data in YARR software (RD53B data processor and decoding tool)
- Explore scans based on PToT data
- Pixel address not from data, but rather from user input (mask staging)
- Tricky to integrate into YARR library. Plan to first implement as a standalone macro


## Backup

| Injection width ( 160 MHz ) | 16-bit data | 11-bit PToT |
| :---: | :---: | :---: |
| 2 | 0111000000000100 | 001110000000 |
| 3 | 1011000000000100 | 010110000000 |
| 4 | 1111000000000100 | 011110000000 |
| 5 | 0011000100000100 | 000110001000 |
| 6 | 0111000100000100 | 001110001000 |
| 7 | 1011000100000100 | 010110001000 |
| 8 | 1111000100000100 | 011110001000 |
| 9 | 0011001000000100 | 000110010000 |
| 10 | 0111001000000100 | 001110010000 |
| 11 | 1011001000000100 | 010110010000 |
| 12 | 1111001000000100 | 011110010000 |
| 13 | 0011001100000100 | 000110011000 |
| 14 | 0111001100000100 | 001110011000 |
| 15 | 1011001100000100 | 010110011000 |
| 16 | 1111001100000100 | 011110011000 |
| 17 | 0011010000000100 | 000110100000 |
| 18 | 0111010000000100 | 001110100000 |
| 19 | 1011010000000100 | 010110100000 |
| 20 | 1111010000000100 | 011110100000 |
| 21 | 0011010100000100 | 000110101000 |
| 22 | 0111010100000100 | 001110101000 |
| 23 | 1011010100000100 | 010110101000 |
| 24 | 1111010100000100 | 011110101000 |
| 25 | 0011011000000100 | 000110110000 |
| 26 | 0111011000000100 | 001110110000 |
| 27 | 1011011000000100 | 010110110000 |
| 28 | 1111011000000100 | 011110110000 |
| 29 | 0011011100000100 | 000110111000 |

