PHENIX FPHX Testing Program

# Smoke Test

1. Power on.
2. Check current. Analog Vdd should be between 0 and 10mA. Digital Vdd should be between 0 and 20mA.

# Reset Test

1. Assert asyncReset.
2. Check current. Analog Vdd should be between 5mA and 9mA (7mA expected). Digital Vdd should be between 11mA and 15mA (13mA expected).
3. Record the current for later analysis.

# Clock Test

1. Inputs inactive.
2. BCO inactive (not strictly necessary)
3. outClk active
4. The Sync Word (19 zeros followed by a 1) should start coming out of serialOut1 and serialOut2.
5. Can we check the LVDS levels here (Vlo=1.0, Vhi=1.4, Vmid=1.2)? Should we?

**COMMENTARY:** At this point, we will have established that the chip is alive and that a fair portion of its digital circuitry is working. Its LVDS drivers and receivers work and we are reaching far enough into the chip to tickle the FIFO serializer a bit. We have no idea yet about the slow control registers, the hit capture logic, the phase control logic or any of the front end circuitry. However, if anything fails Tests 1, 2, or 3, it is clearly RED and must be discarded.

# Register Tests Part 1 – everything but Kill/Inject (Register 1) and LVDS Current (Register 17)

1. Al already has a well defined, successful register test. Use that one without modification.

# Register Tests Part 2 – LVDS Current (Register 17)

1. Write 00001111b to Register 17.
2. Read it back.
3. Check the LVDS levels of the slowControlOut.
4. Write 00011110b to Register 17.
5. Read it back
6. Check the LVDS levels. They should not have changed significantly.
7. Write 00111100 to Register 17.
8. Read it back.
9. Check the LVDS levels. They should not have changed significantly.
10. Write 01111000 to Register 17.
11. Read it back.
12. Check the LVDS levels. They should not have changed significantly.
13. Write 11110000 to Register 17.
14. Read it back.
15. Check the LVDS levels. They should not have changed significantly.

**COMMENTARY:** The point is that for each 1 in Register 17, a fixed number of milliamps are driven out of the LVDS drivers. By keeping the number of 1s a constant, we should be keeping the LVDS levels a constant. If the LVDS levels are changing, then one or more bits in Register 17 is flawed.

# Register Tests Part 3 – Kill/Inject (Register 1)

1. Frankly, it is not possible to test the Kill/Inject Register independently of the Channels themselves.

# Register Tests Part 4 – Defaults

1. Hit Reset
2. Read every register

|  |  |  |
| --- | --- | --- |
| Register Address | Contents | Default Value |
| 1 | Mask | N/A |
| 2 | Digital Control | 01 |
| 3 | Vref | 01 |
| 4 | ThDAC 0 | 08 |
| 5 | ThDAC 1 | 16 |
| 6 | ThDAC 2 | 32 |
| 7 | ThDAC 3 | 48 |
| 8 | ThDAC 4 | 80 |
| 9 | ThDAC 5 | 112 |
| 10 | ThDAC 6 | 144 |
| 11 | ThDAC 7 | 176 |
| 12 | N2Sel/N1Sel | 70 |
| 13 | LeakSel/FB1Sel | 04 |
| 14 | P2Sel/P3Sel | 64 |
| 15 | BWSel/GSel | 33 |
| 16 | InjSel/P1Sel | 05 |
| 17 | LVDS Current | 15 |
| 18 | Resets | N/A |

# Charge Injection Test Part 1 – On-board Pulser

**COMMENTARY:** The DCAL scheme for Hit/No Hit testing of each channel should be sufficient. I think that keeping the time-per-chip to a minimum is probably more important than an exhaustive testing at this point.

# Charge Injection Test Part 2 – External Pulser

**COMMENTARY:**  By this time, we now know that the chip is working and its registers function. All that is left is to see if it’s Analog-to-Digital Converters are working. To that end, I think it is important that in some way we make each channel turn on and then output 0 to 7 on its ADC bits. The first idea that comes to mind is to turn on one channel with all the others off, and start pulsing. After each pulse, raise the external pulser by 1 count and pulse again. Record the count value at which each channel switched from ADC(X) to ADC(X+1). If a channel doesn’t show 0 through 7, the chip is RED and should be discarded. Unfortunately, this algorithm sounds really time consuming to me, so I think Al should be given the freedom to alter this test to make it more efficient.

# “The Drake”

1. Turn on a single channel.
2. Set All Thresholds (ThDACn) to maximum.
3. Set the external pulser charge to Value A (to be determined).
4. Pulse the channel 100 times. Record the number of hits and the number of no hits. If there are any hits, record the ADC value.
5. Decrease the Thresholds by 1 count.
6. Repeat the loop from D to F until the thresholds are 0. Record the point at which 50% of the channels fire and the width of the step up from 0% of channels firing and 100% of channels firing.
7. Repeat the loop from B to F with the external pulser charge set to Value B (to be determined).
8. Repeat the loop from B to F with the external pulser charge set to Value C (to be determined).
9. Alter the Gain value (Register 15, lower 3 bits) to 000. Repeat Loop B to H.
10. Alter the Gain value to 010. Repeat Loop B to H.
11. Alter the Gain value to 100. Repeat Loop B to H.
12. Turn off the current channel and turn on the next channel.
13. Repeat loop from B to L until all 128 channels have been tested.

**COMMENTARY:**  The Drake is not meant to be used on every chip on every wafer - not unless something has gone dramatically wrong. That would take much, much too long. However, doing the Drake on several chips per wafer as a pre-test to determined the general ballpark values we can expect from each wafer is a good idea. I would suggest 4 chips in each the lower-left, upper-left, center, upper-right, and lower-right.