nsCamera Timing Tutorial

nsCamera 2.1.2

Jeremy Martin Hill

LLNL-CODE-838080

High-Speed Timing

- `CA.setTiming('AB', (4,2), delay=3)`
  - Hemisphere selection: ‘AB’ (both hemispheres)
  - Tuple sets timing
    - Integration (‘On’) = 4 ns
    - Interframe (‘Off’) = 2 ns
  - Initial delay = 3 ns

- Rules for exact timing (You Get What You Ask For):

Icarus: $\left(\text{Delay} + (4 \times \text{On}) + (3 \times \text{Off})\right) \leq 40$

Daedalus: $\left(\text{Delay} + (3 \times \text{On}) + (2 \times \text{Off})\right) \leq 40$

OR

$$\text{Delay}=0 \textit{ AND } \text{On}+\text{Off} = (20 \text{ or } 40)$$

Arbitrary High-Speed Timing

- `CA.setArbTiming('AB', [1,2,3,4,5,6])  // Daedalus timing`

- List sets timing
  - Initial delay = 1
  - Frame 0 integration / interframe = 2/3
  - Frame 1 integration / interframe = 4/5
  - Frame 2 integration / interframe = 6

- Rule for exact timing (You Get What You Ask For):

Sum of list elements ≤ 40

Manual Timing

- Manual timing allows you to set integration and interframe intervals in multiples of 25 ns up to a total of 2^{30} counts, or approximately 27 seconds.

- CA.setManualTiming([100,50,100,50,100,50,100])
  - F0on, F0off, F1off, F2on, F2off, F3on, F3off, F4on (latter two for Icarus only)
  - If five/seven values are given in the list, timing is set the same for both hemispheres
  - If ten/fourteen values are given, the first five/seven are assigned to hemisphere A, the rest to hemisphere B
  - Values are rounded down to nearest multiple of 25 ns (if you ask for 140 ns, you'll get 125 ns)

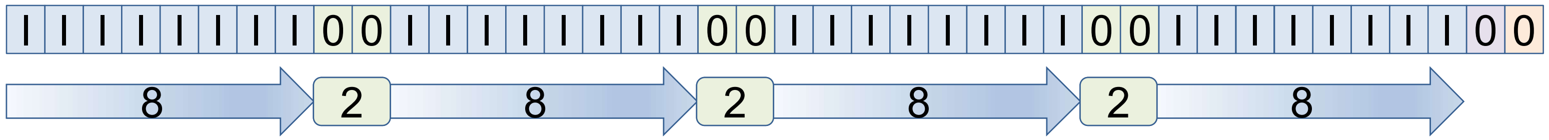
Get What You Ask For

- If you will always keep within the rules listed on the previous slides, you’re done
  - You can go back to the lab now

- If you’re willing to break a few rules, you want to use more advanced features, or you’re just curious about how the sausage is made, keep reading…

Sensor timing register (High Speed Timing)

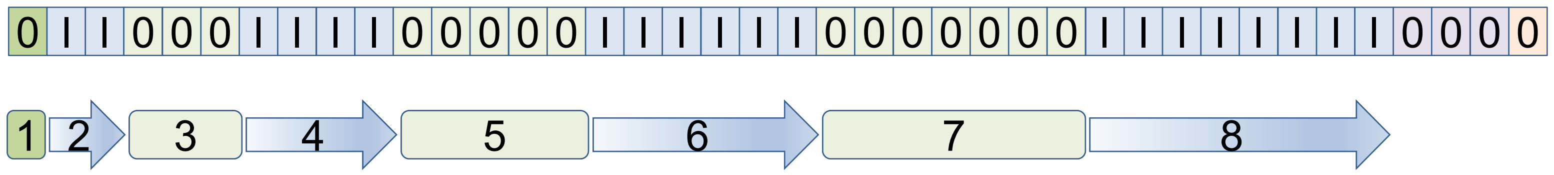
- Acquisition timing is controlled by a 40-bit register on the sensor. Each bit represents one nanosecond of timing. The final* bit is fixed to zero 0
- The sensor rolls through the register one bit at a time. Initial zeroes 0 indicate a delay; ones I indicates integration; subsequent zeroes 0 are interframe breaks.
  - The sensor continues through the register until all frames have been acquired
  - If the register runs out before all frames have been acquired, the sensor returns to the beginning of the register and continues until all frames have been acquired
  - The software will fit as many copies of the timing as possible into the register; unassigned bits are left as zeroes. 0
- Example: 8/2 timing on Icarus:


I I I I I I I I 0 0 I I I I I I I I 0 0 I I I I I I I I 0 0 I I I I I I I I 0 0
8
2
8
2
8
2
8

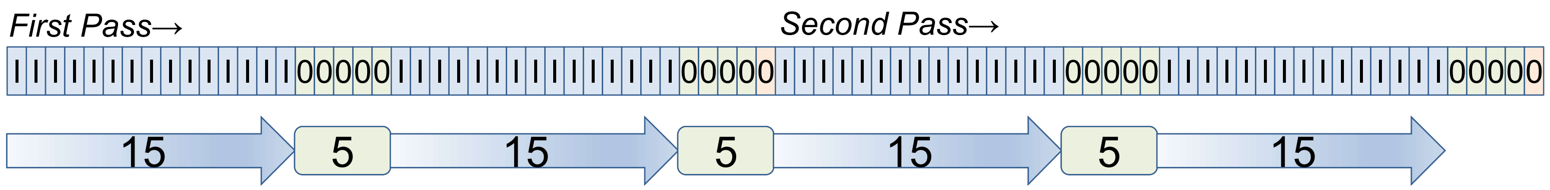
* This isn't exactly how the register is implemented physically, but this is the representation used in the software and the FPGA board and is slightly easier to understand. See the official UXI documentation for details.

Expected timings (when you follow the rules)

- Timing (regular or arbitrary) fits inside 40 ns, e.g. [1,2,3,4,5,6,7,8] on Icarus2:

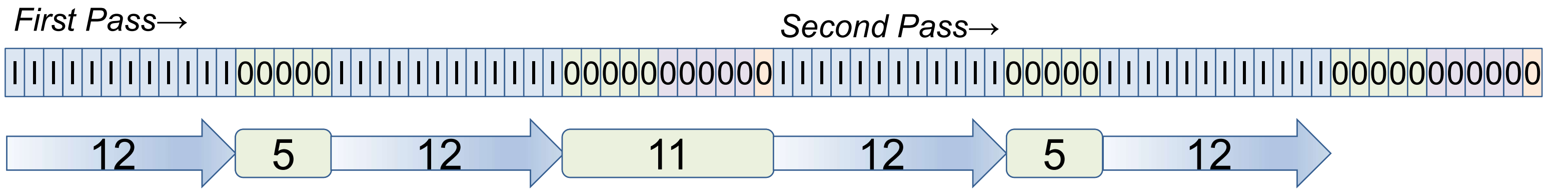

0 I I 0 0 0 I I I I 0 0 0 0 0 I I I I I I 0 0 0 0 0 0 0 I I I I I I I I 0 0 0 0
1
2
3
4
5
6
7
8

- Delay = 0 and On+Off = 20 or 40, e.g., 15/5 on Icarus2:


First Pass→
Second Pass→
IIIIIIIIIIIIIII00000IIIIIIIIIIIIIII00000IIIIIIIIIIIIIII00000IIIIIIIIIIIIIII00000
15
5
15
5
15
5
15

When you don’t follow the rules, odd things can happen

- 12/5 timing on Icarus2
  - The software will fit as many copies of the on/off sequence into the 40 bits as it can; in this case, it fits twice $(2 \times (12 + 5) = 34)$; the remaining bits are left as zeroes
  - The result is an unasked-for extension of the interframe between frames 1 and 2

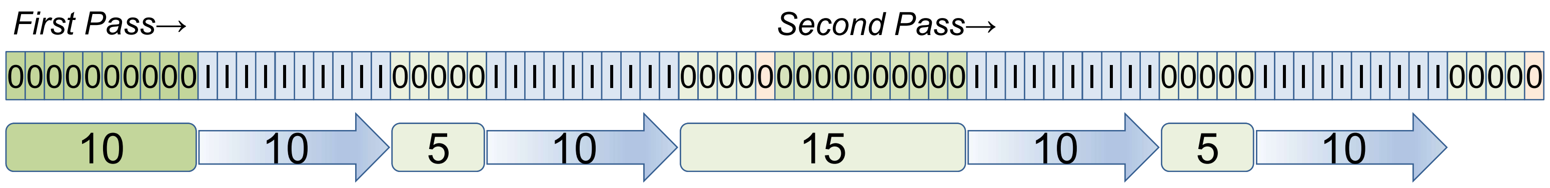

First Pass→
Second Pass→
IIIIIIIIIIII00000IIIIIIIIIIII00000000000IIIIIIIIIIII00000IIIIIIIIIIII00000000000
12
5
12
11
12
5
12

  - The nsCamera software will warn you if this is going to happen:

```
WARNING: [Icarus2] setTiming: Due to sequence length, the actual
timing sequence for side A will be {0} [12, 5, 12, 11, 12, 5, 12]
```

Initial delays recur if the register requires multiple passes

- {10} 10/5 timing on Icarus2
  - This timing 'fits' into 40 bits without adding any extra zeroes at the end, but the delay introduces an unexpectedly extended interframe



First Pass→
Second Pass→
0000000000IIIIIIIIII00000IIIIIIIIII000000000000000IIIIIIIIII00000IIIIIIIIII00000
10
10
5
10
15
10
5
10

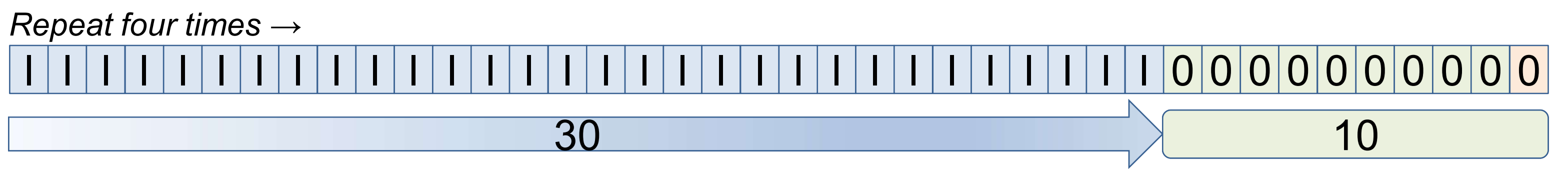
  - Again, the software will provide a warning:

```
WARNING: [Icarus2] setTiming: Due to sequence length, the actual
timing sequence for side A will be {10} [10, 5, 10, 15, 10, 5, 10]
```

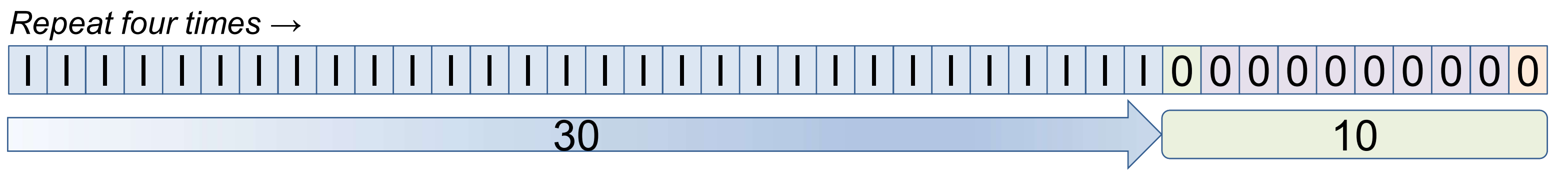
Any timing request that fits only once into the 40-bit register will result in N/(40-N) timing

- Example: Requests for 30/1 or 30/10 timings will produce the same register setting, and the actual timing will necessarily be 30/10:

- 30/10:

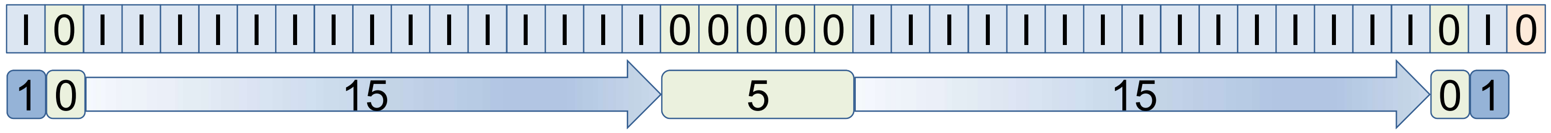

Repeat four times →
I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I 0 0 0 0 0 0 0 0 0 0
30
10

- 30/1:


Repeat four times →
I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I 0 0 0 0 0 0 0 0 0 0
30
10

Icarus1 sensors will always have an additional ‘delay’ introduced by Frame 0 being skipped

- Icarus1 sensors still program four frames into the register, so the Rules still apply
  - For best results, if assigning all four frames with conventional timing wouldn’t fit into the 40-bit limit, try arbitrary timing to minimize the unacquired frames
  - Example: 15/5 timing using `CA.setTiming('AB',(15,5))` would introduce spurious pauses that you can avoid by using `CA.setArbTiming 'AB', [0,1,1,15,5,15,1,1])`



I 0 I I I I I I I I I I I I I I I 0 0 0 0 0 I I I I I I I I I I I I I I I 0 I 0
1
0
15
5
15
0
1

- The nsCamera software will remind you that there is a ‘bonus’ delay introduced by Frame 0:

```
WARNING: [Icarus] Due to use of the Icarus model 1 sensor, the
actual timing sequence for side A will be {2} [15, 5, 15]
```

High Full-Well Timing (Daedalus only)

- High Full-Well Timing utilizes all three shutter systems on the sensor to capture a single frame
  - Program the desired length of the frame, with an interframe of 1
  - `CA.setHighFullWell(True)`
  - `CA.setTiming ('AB', (35,1), delay=3)`
- Maximum timing: $(\text{Delay} + \text{On} + \text{Off}) \leq 40$
  - The software may complain about unexpected timings, but only the Frame 0 integration time is relevant

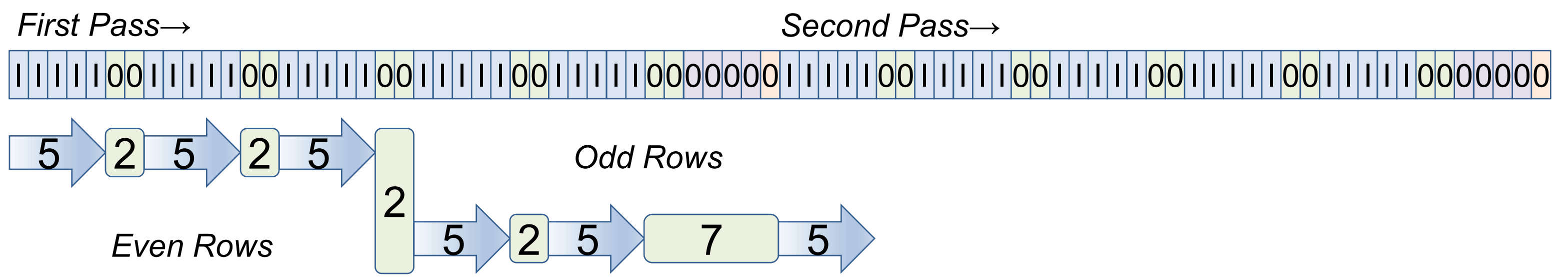
Row Interlacing Timing (Daedalus only)

- Row Interlacing allows for collection of multiples of the three frames available to Daedalus by collecting interlaced subsets of the image.
  - Example: single-row interlacing (use `CA.setInterlacing(1)`) collects three frames of only the even-numbered rows, and then three frames of the odd-numbered rows
    - Each subimage is 512x512
  - Interlacing is available from 1 to 1023 rows (where each row is acquired individually)
  - The nsCamera software automatically fills the timing register with as many copies of the requested timing, in anticipation of possible interlacing

- WARNING: the timing register will keep rolling until all frames are acquired. Timing settings that work accurately for three frames may not for multiples of three
  - The nsCamera software does not yet monitor this issue for interlacing

Row Interlacing, continued

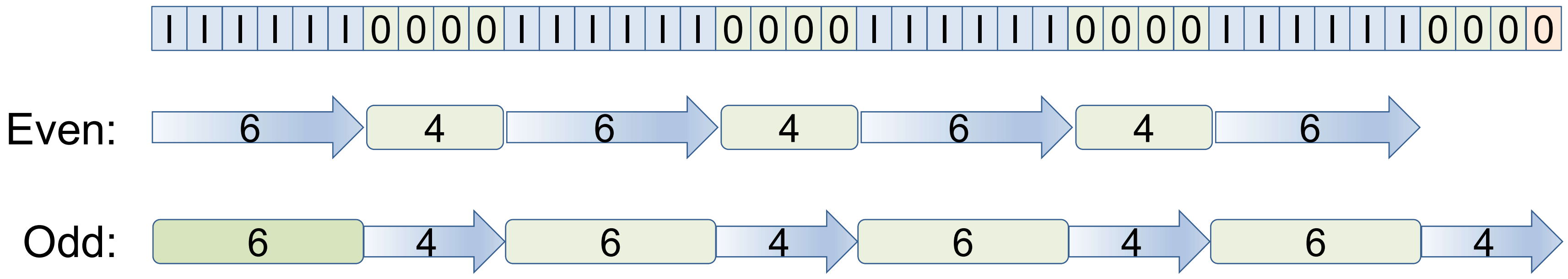
- 5/2 timing works fine without interlacing, but introduces an anomalous pause when used for interlacing:



- Use `CA.setInterlacing(0))` to go back to normal (uninterlaced) operation

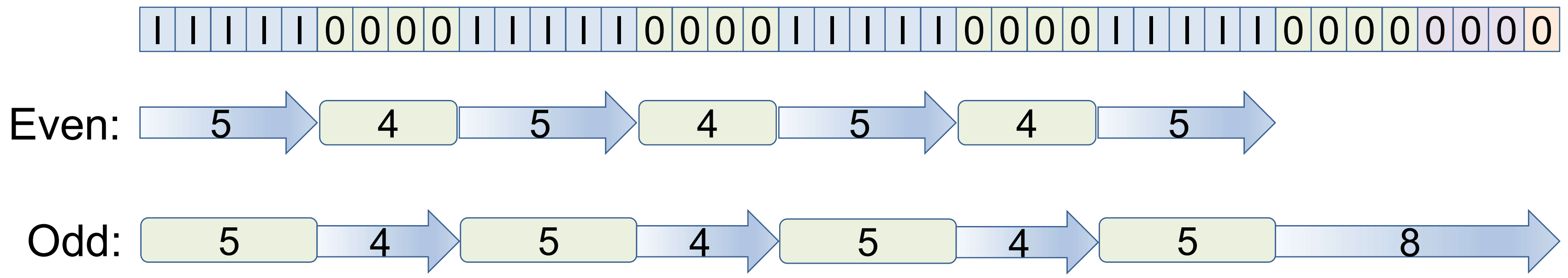
Zero Dead Timing

- Zero Dead Timing is a particular interlacing scheme whereby the even rows are acquired as normal according to the high-speed timing setting, while the odd rows are acquired during the interframes
  - Acquisition is constant (no ‘dead’ time) until all frames are acquired
  - `CA.setZeroDeadTime(True)`
  - Example: 6/4 timing on Icarus2:


I I I I I I 0 0 0 0 I I I I I I 0 0 0 0 I I I I I I 0 0 0 0 I I I I I I 0 0 0 0
Even:
6
4
6
4
6
4
6
Odd:
6
4
6
4
6
4
6
4

Zero Dead Timing, continued

- As with the regular interlacing, you must be watchful for actual timings that you're not expecting:
  - Example: 5/4 timing on Icarus2:



I I I I I 0 0 0 0 I I I I I 0 0 0 0 I I I I I 0 0 0 0 I I I I I 0 0 0 0 0 0 0 0
Even:
5
4
5
4
5
4
5
Odd:
5
4
5
4
5
4
5
8