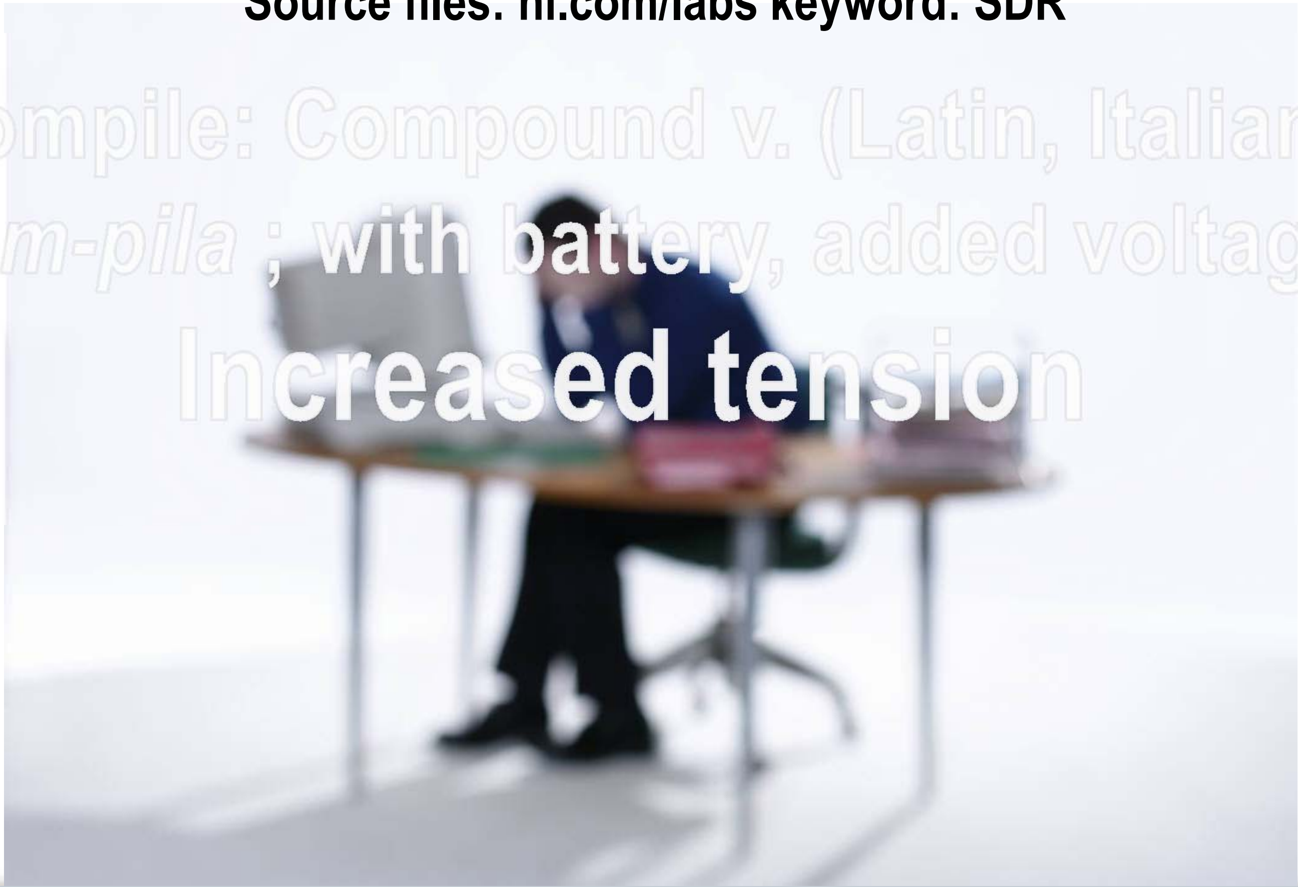
Using a graphical interface for Fast FPGA design revision in SDR hierarchical structure

Frank Raffaeli: Principal RF Engineer, National Instruments

Source files: ni.com/labs keyword: SDR



Compile: Compound v. (Latin, Italian)

com-pila ; with battery, added voltage

Increased tension

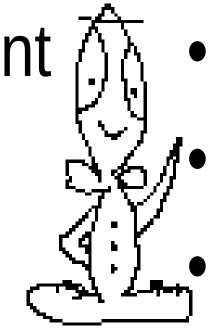
FPGA Signal Processing

- Numerically fast, high bandwidth
- Portable
- Deployable
- Re-configurable
- Slow design / prototyping turn-around
- Bottleneck through DSP engineer

Re-Configurability – What it is and isn’t

IS

- Fast re-configuration / deployment
- Existing, mainstream Tools
- A firmware / software Macro
- Xilinx HW + LabViewFPGA + sauce
- Simple with penalties
- *low-cost*



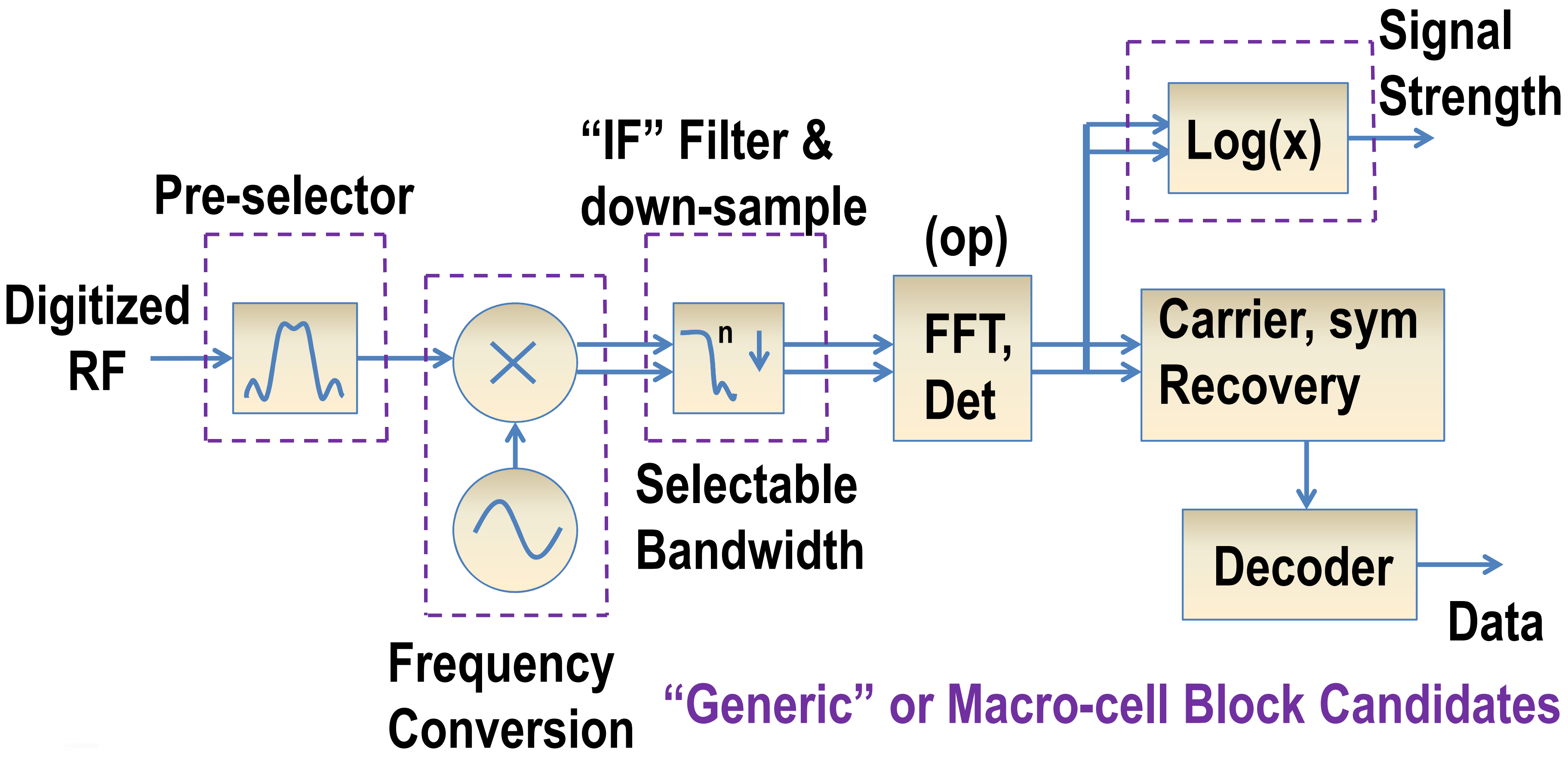
ISN’T

- Partial Re-config.
- Fast Xilinx compile
- A hardware Macro
- Custom HW, Software
- Complex and Cryptic

Aka: RTC, RCM, Macro-cell

B.H.A.G. Receiver Physical Layer in FPGA

Digital Proc (NI-7965R with V5 SX-95T)



Digitized RF

Pre-selector

Frequency Conversion

“IF” Filter & down-sample

Selectable Bandwidth

(op)

FFT, Det

Log(x)

Signal Strength

Carrier, sym Recovery

Decoder

Data

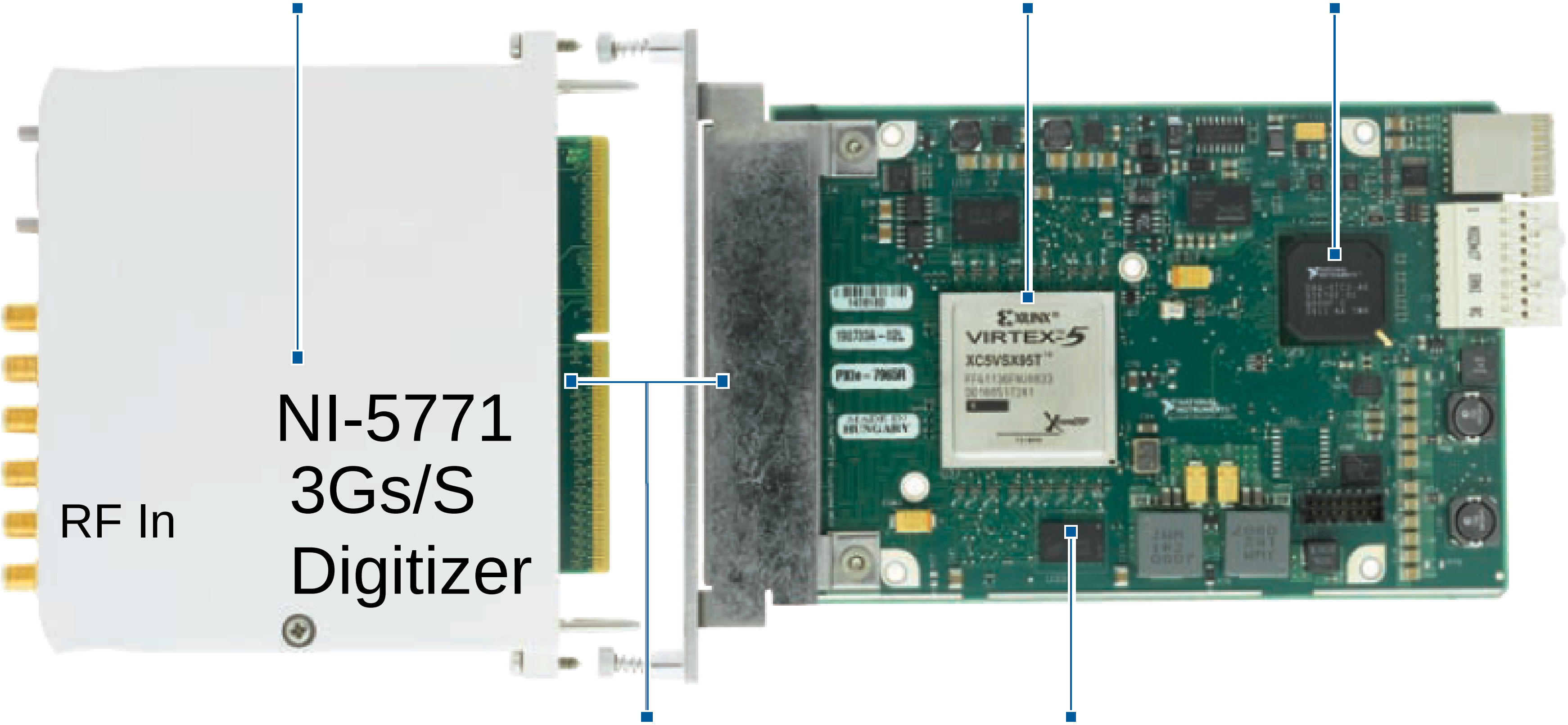
“Generic” or Macro-cell Block Candidates

B.H.A.G. Radio – Run-time configurable?

NI FlexRIO Adapter Module
(NI, Third-Party, or Custom)

Xilinx
Virtex-5 FPGA

PXI or PXI Express
Interface



RF In

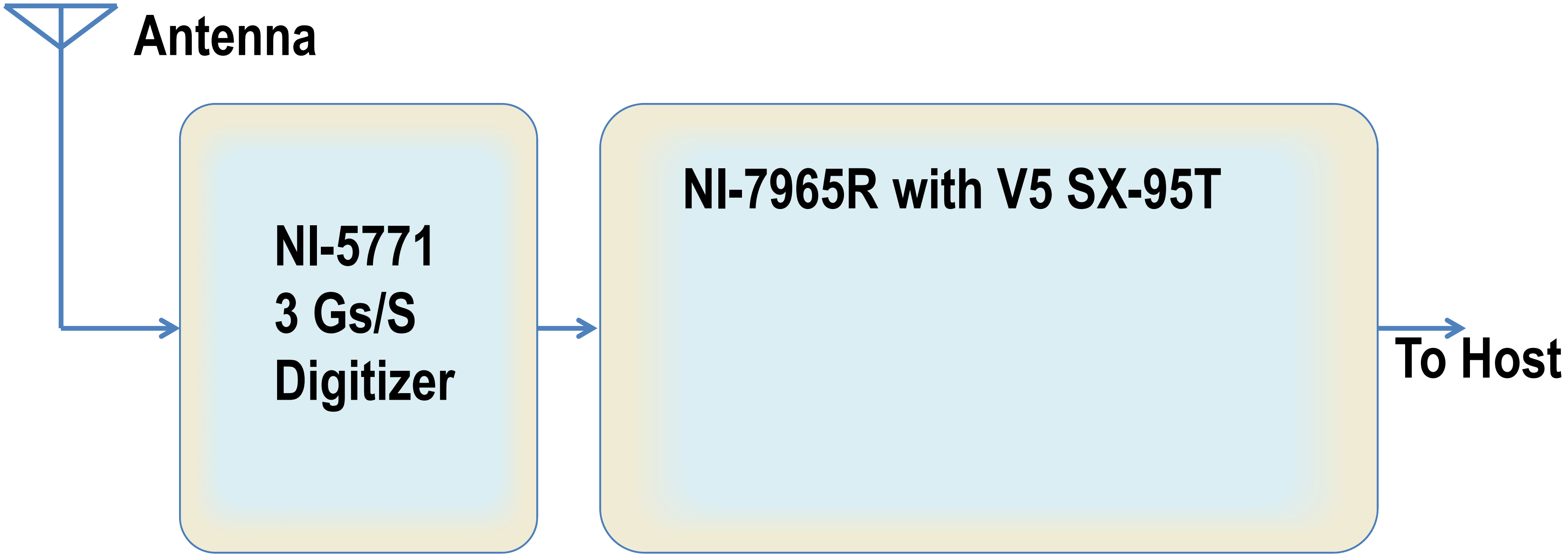
NI-5771
3Gs/S
Digitizer

Adapter Module Interface

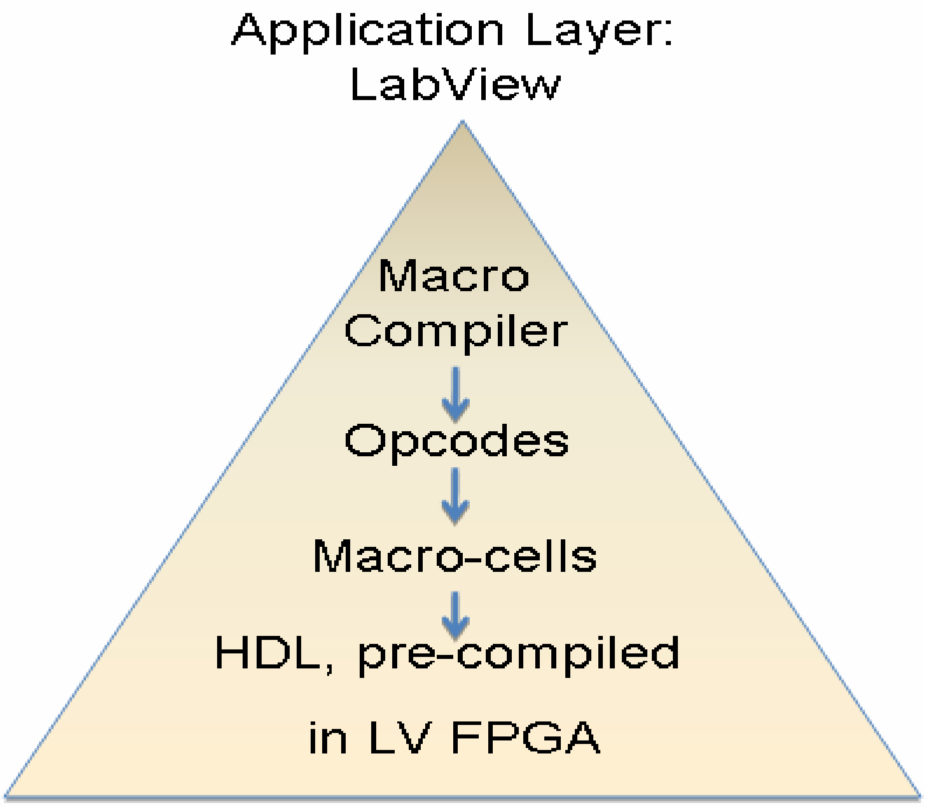
Onboard DRAM

Figure 1. NI FlexRIO Architecture

Demo – Re-configurable Receiver based on Xilinx SX95T (Virtex-5)



Soft Macrocell support pyramid



Application Layer:
LabView

Macro
Compiler
↓
Opcodes
↓
Macro-cells
↓
HDL, pre-compiled

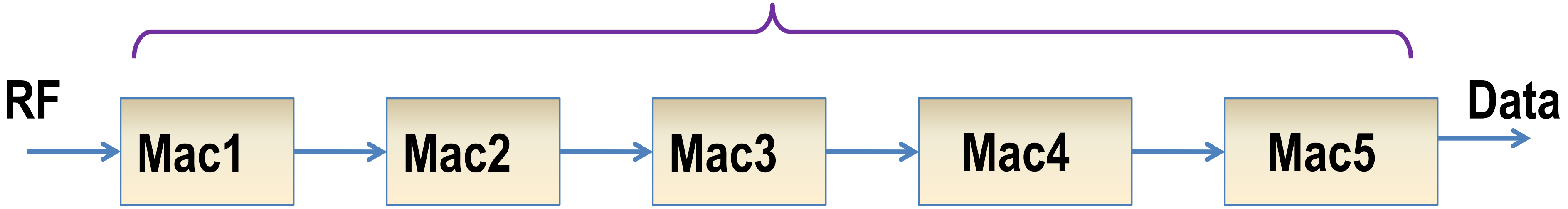
in LV FPGA

Physical layer “Generic”

Soft

What the receiver chain would look like implemented with macro-cells

Function of all blocks are configurable at run-time



“Front End”

Front-End
Pre-selector
Frequency Conversion
Hi speed filtering

“Back End”

Back-End
FFT, Demodulator
Precision filtering
Complex functions

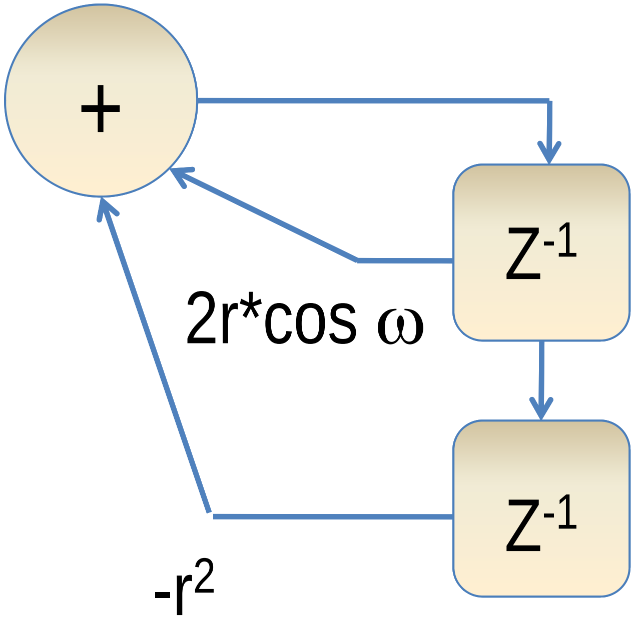
Can we make a generic block function - a.k.a “Soft Macro” ?

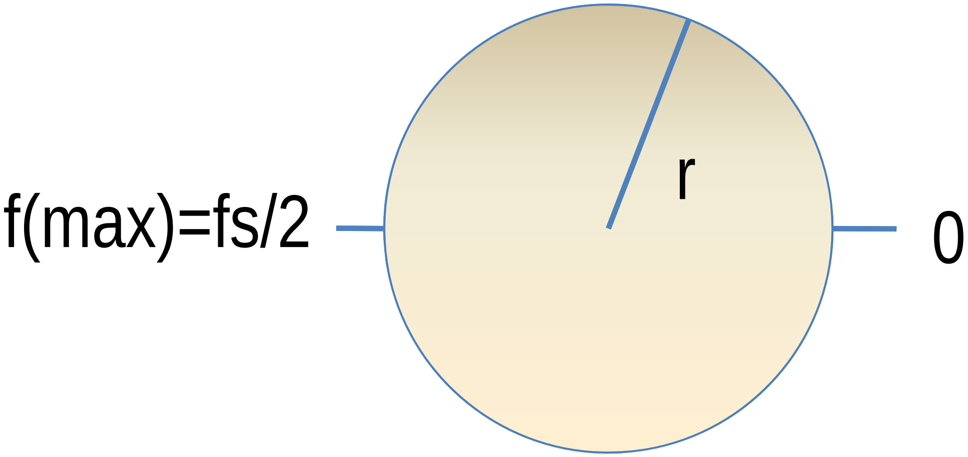
Requirements

- Parameterized
- Configurable at run-time
- Compiler support**
- High performance – what compromises?
- Reasonably efficient utilization

Zoom In on Oscillator Function

Macro compatible?

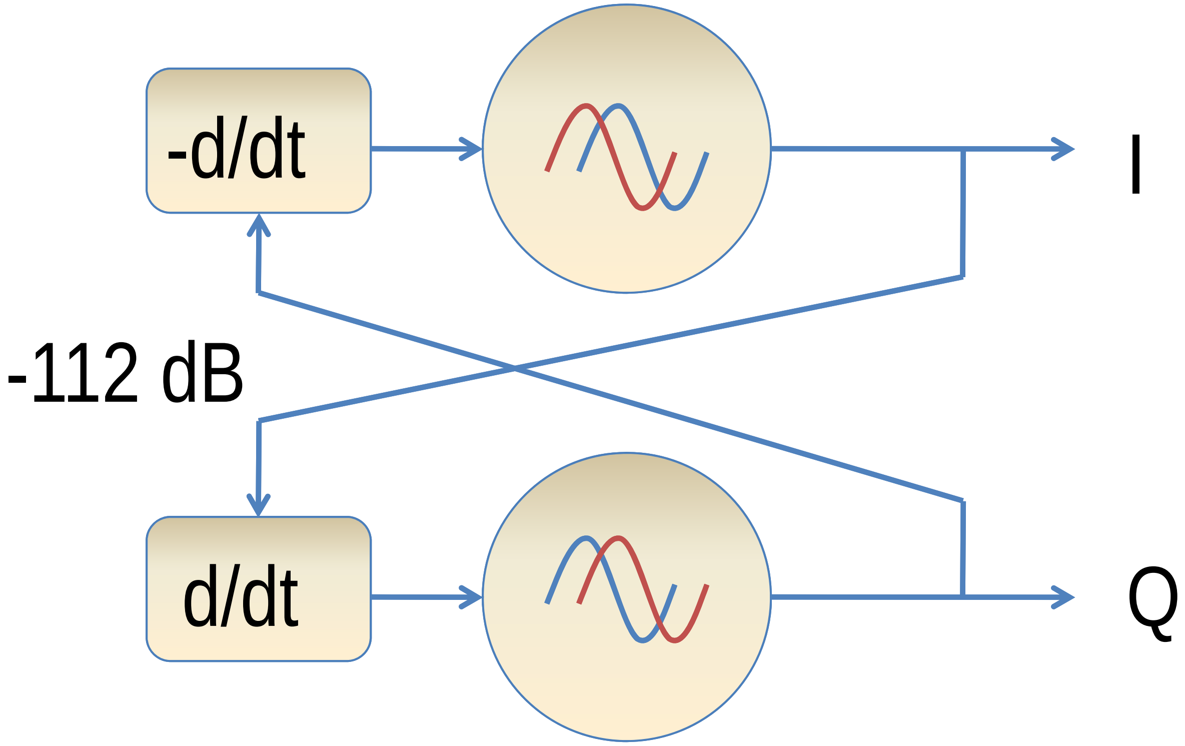




“Z” domain: unit circle

R ≤ 1 for stability

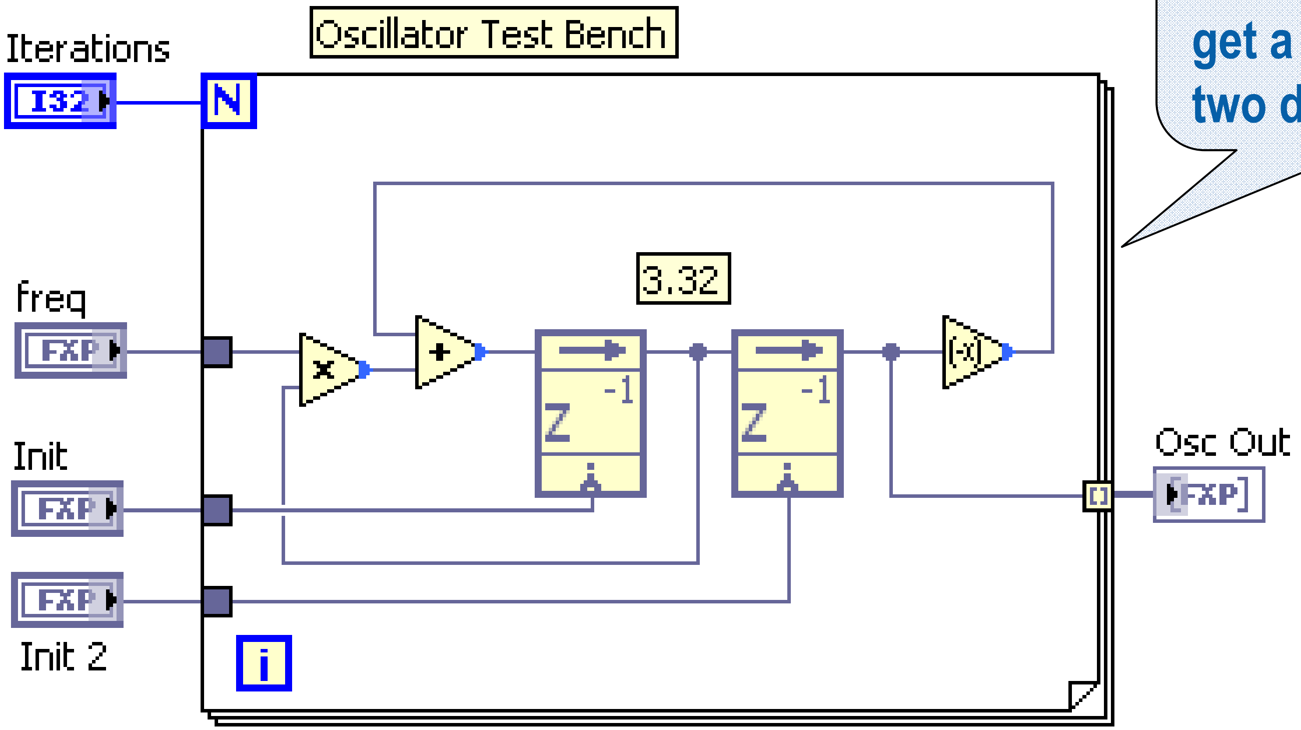
We need I & Q: The Injection Lock trick



Video

–d/dt of cos(x) = sin(x), etc …. Adding a small signal to an oscillator makes the oscillator lock to that signal.

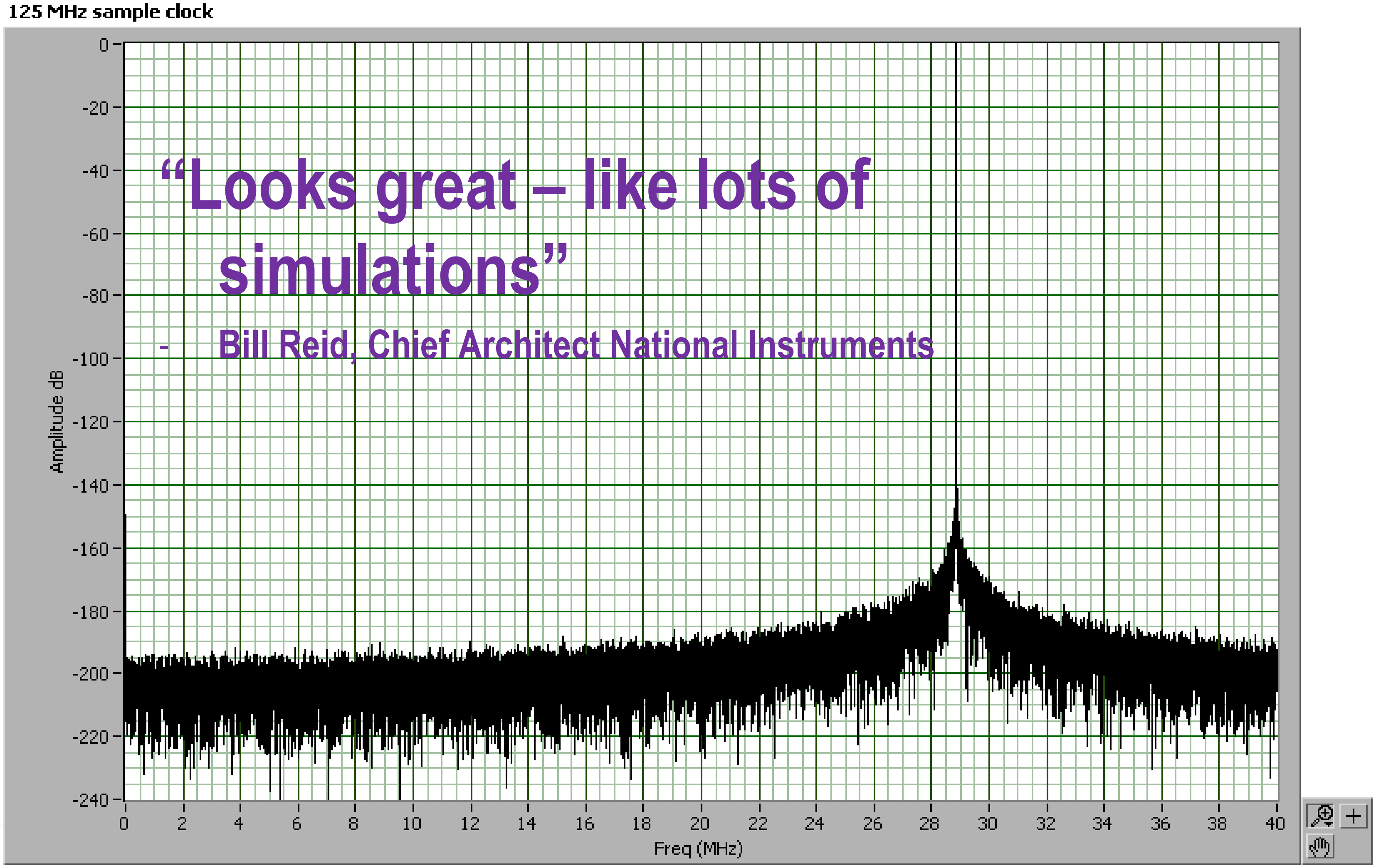
Aha Moment



Surprising (to an analog engineer) to get a sine wave from two delay elements

Is it really a sine wave?

Test-bench result for Oscillator (32-bit)



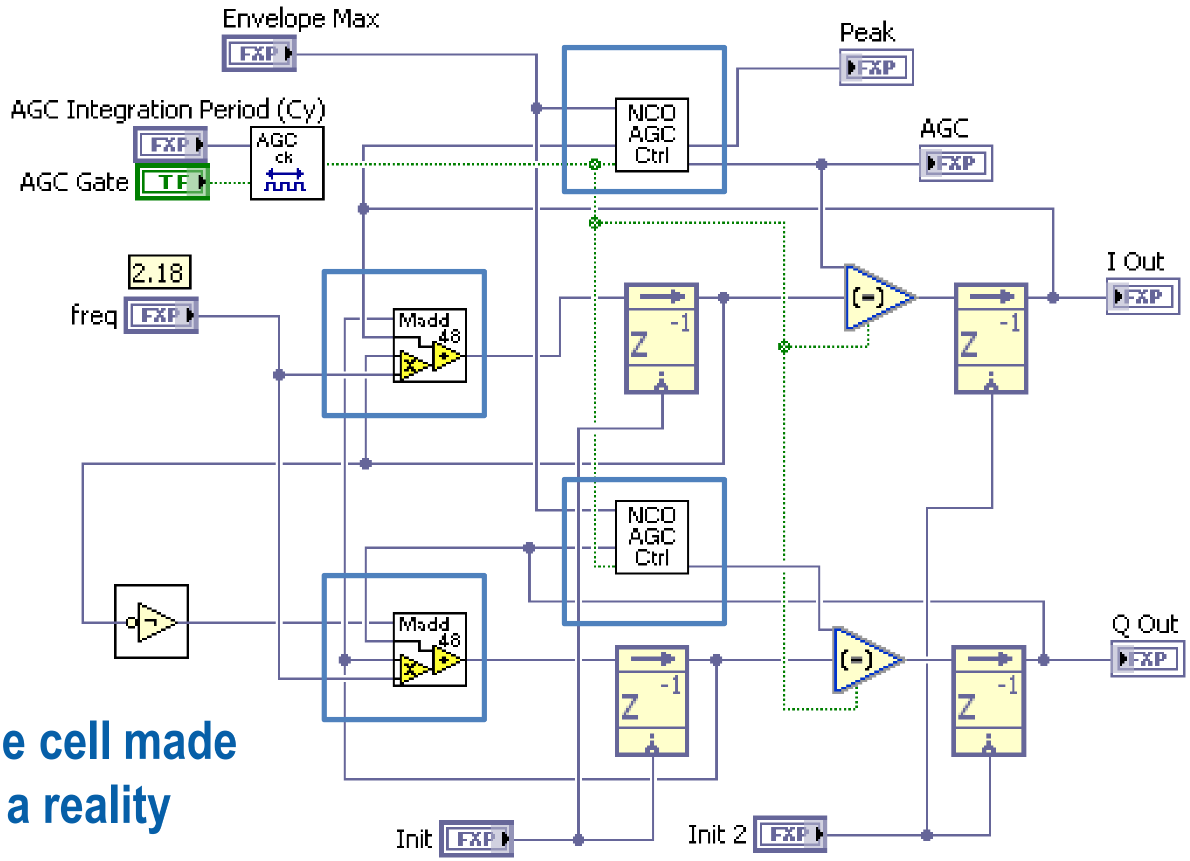
“Looks great – like lots of simulations”

- Bill Reid, Chief Architect National Instruments

Test-bench to reality transition - test bench limitations:

1. Multiply & add function doesn’t meet timing
2. Implementation is large and inefficient
3. See #1 again

Injection-Locked Dual Oscillator - reality

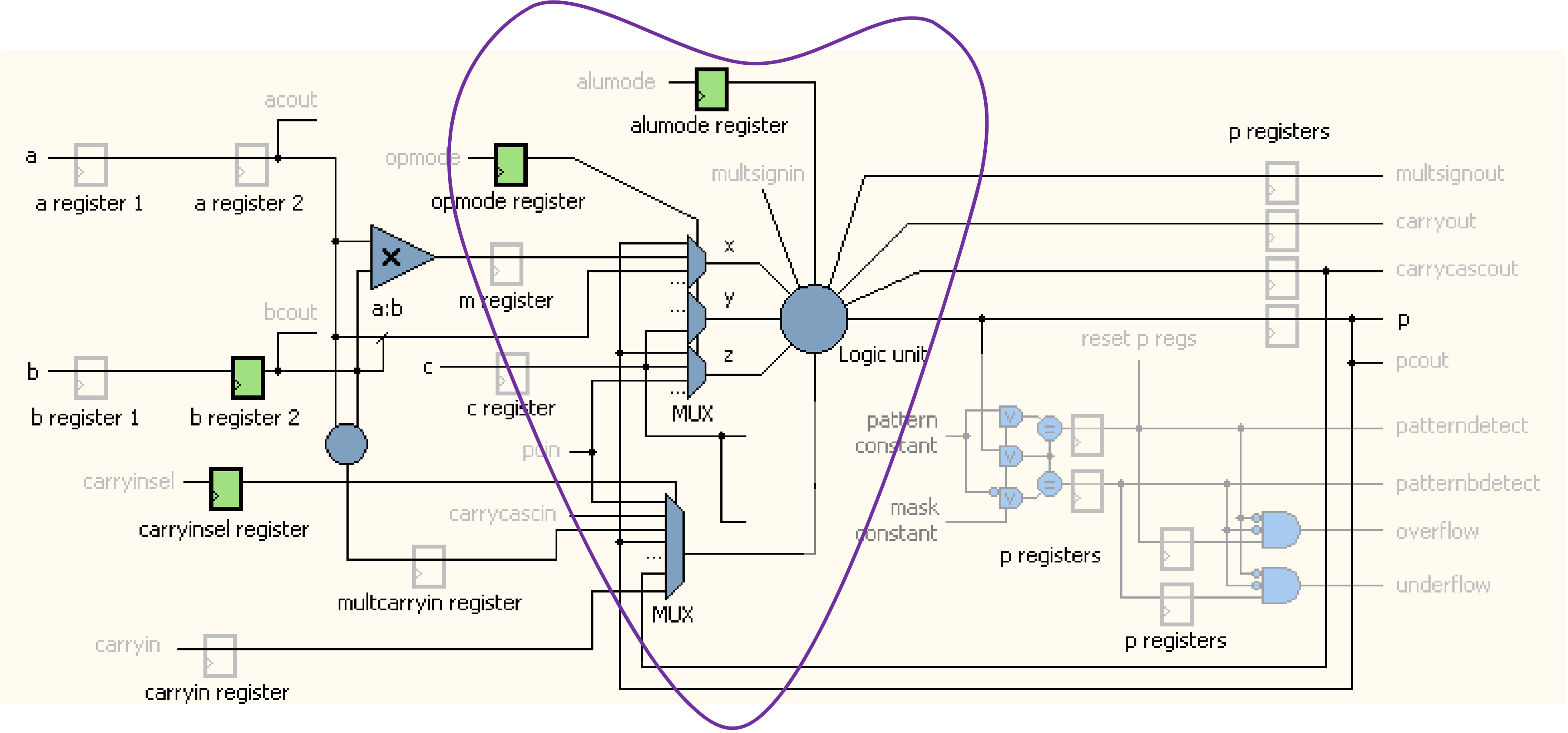


LabView FPGA Module

DSP48e cell made timing a reality

How can this architecture be implemented in a re-configurable macro-cell?

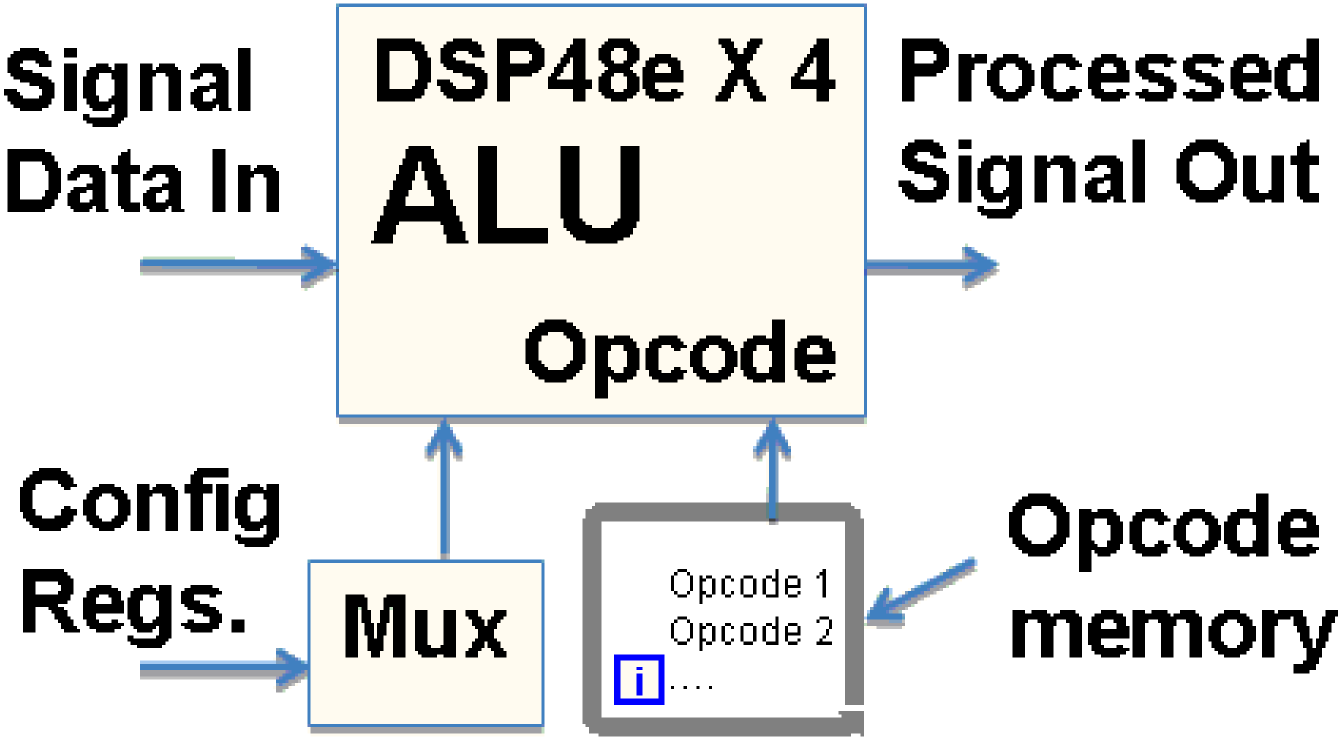
Xilinx DSP48e – already reconfigurable



The heart of the re-configurable engine

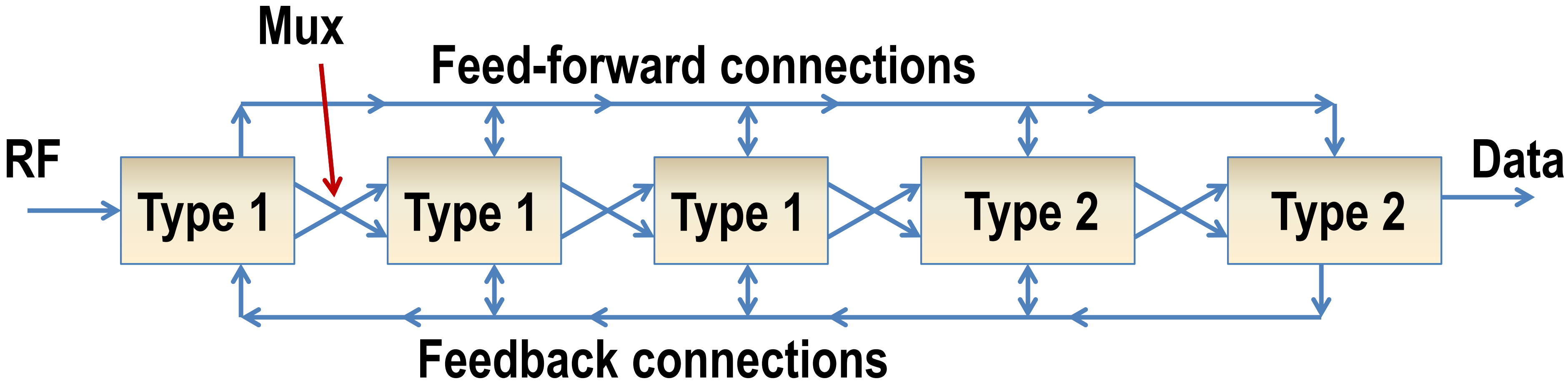
And the main building block of the type 1 macro-cell

Anatomy of the (type 2) Macrocell



Canonical Generic Macro-cells

Receiver Physical Layer – now a pre-compiled structure in the FPGA
“Generic” macro-cell reduced to two types w/ interconnect framework



Type 1 – optimized for:
Fast Filters
Oscillators, multipliers
Hi speed, *static*

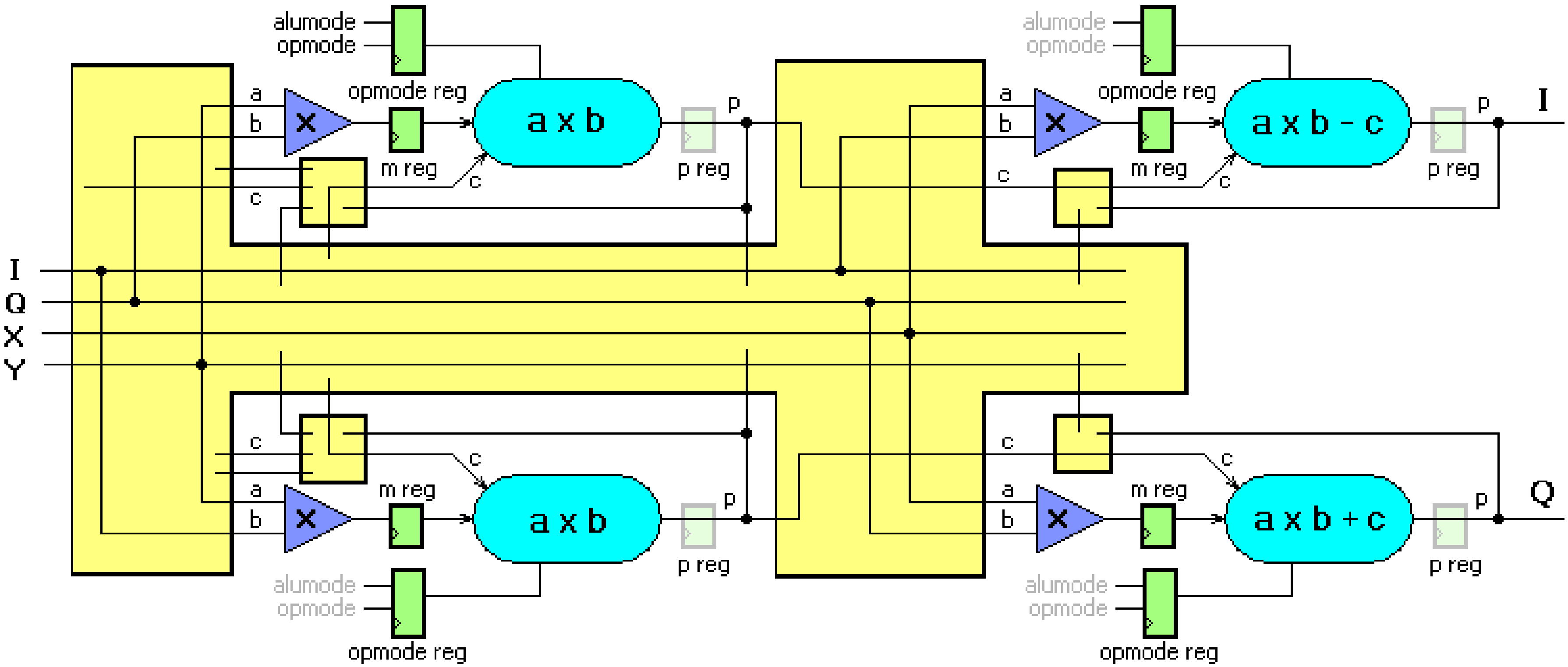
Type 2 – optimized for:
IF & recursive Filters
Demodulators, FFT
Med speed, *dynamic*

Comparing the Oscillator Footprint “Hardwired” vs. Type 1 Soft Macro-Cell

- FPGA Resource Consumption - Courtesy Dan Baker

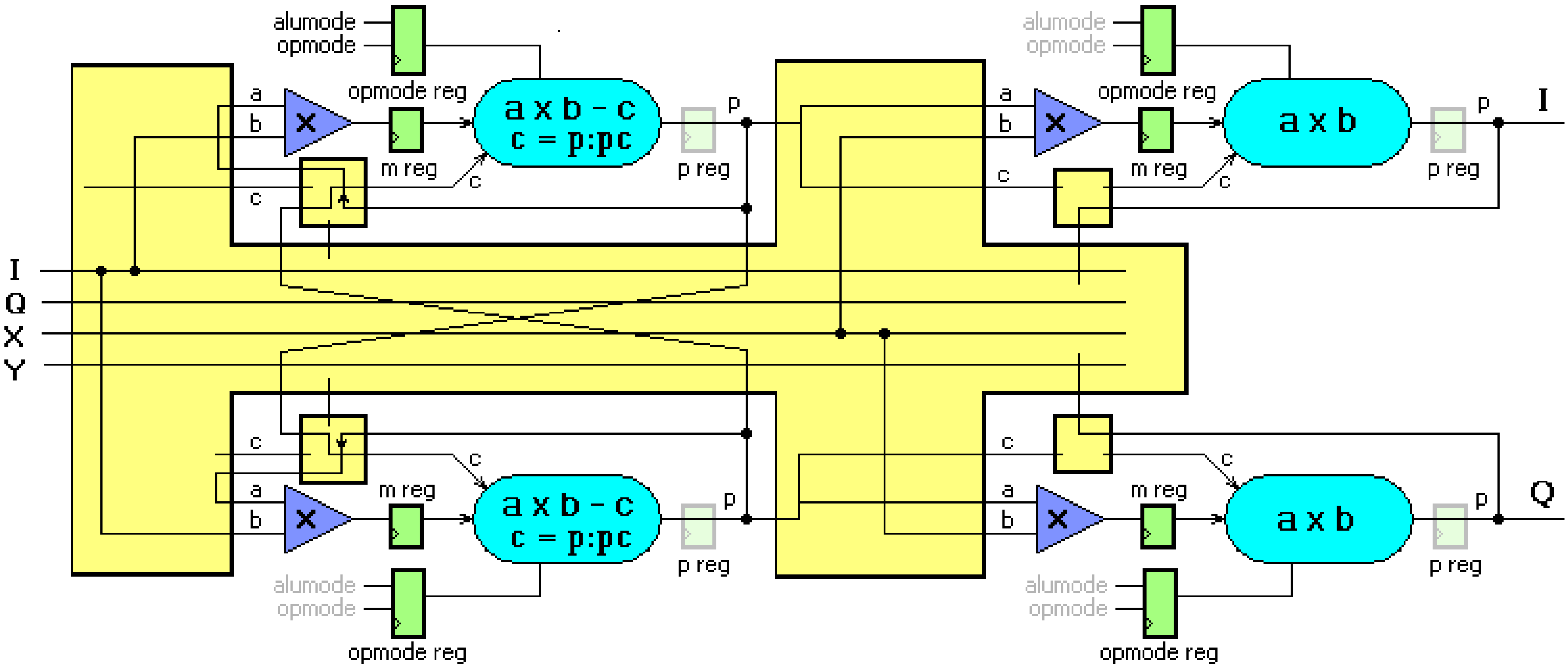
Design	Slice FFs	Slice LUTs	DSP48Es	Block Rams
Dual Oscillator in T1 (18 bit)	340	294	4	0*
Sine & Cosine	1128	1108	0	0
Sine & Cosine with Dither	805	1093	3	0
Coregen DDS Taylor	389	260	1	1
Coregen DDS Dither	482	312	0	1

Complex multiplier in RTC Macrocell



(I+Qi) * (X+Yi) = I*X – Q*Y (real) ; Q*X + I*Y (imaginary)

Oscillator and ACG in RTC Macrocell



Oscillators are cross-coupled to maintain quadrature.

I = frequency, X = AGC control

Implementation efficiency vs. Compile time - the benefits of trial-and-error

"We are ready for any unforeseen event that may or may not occur" - Dan Quayle

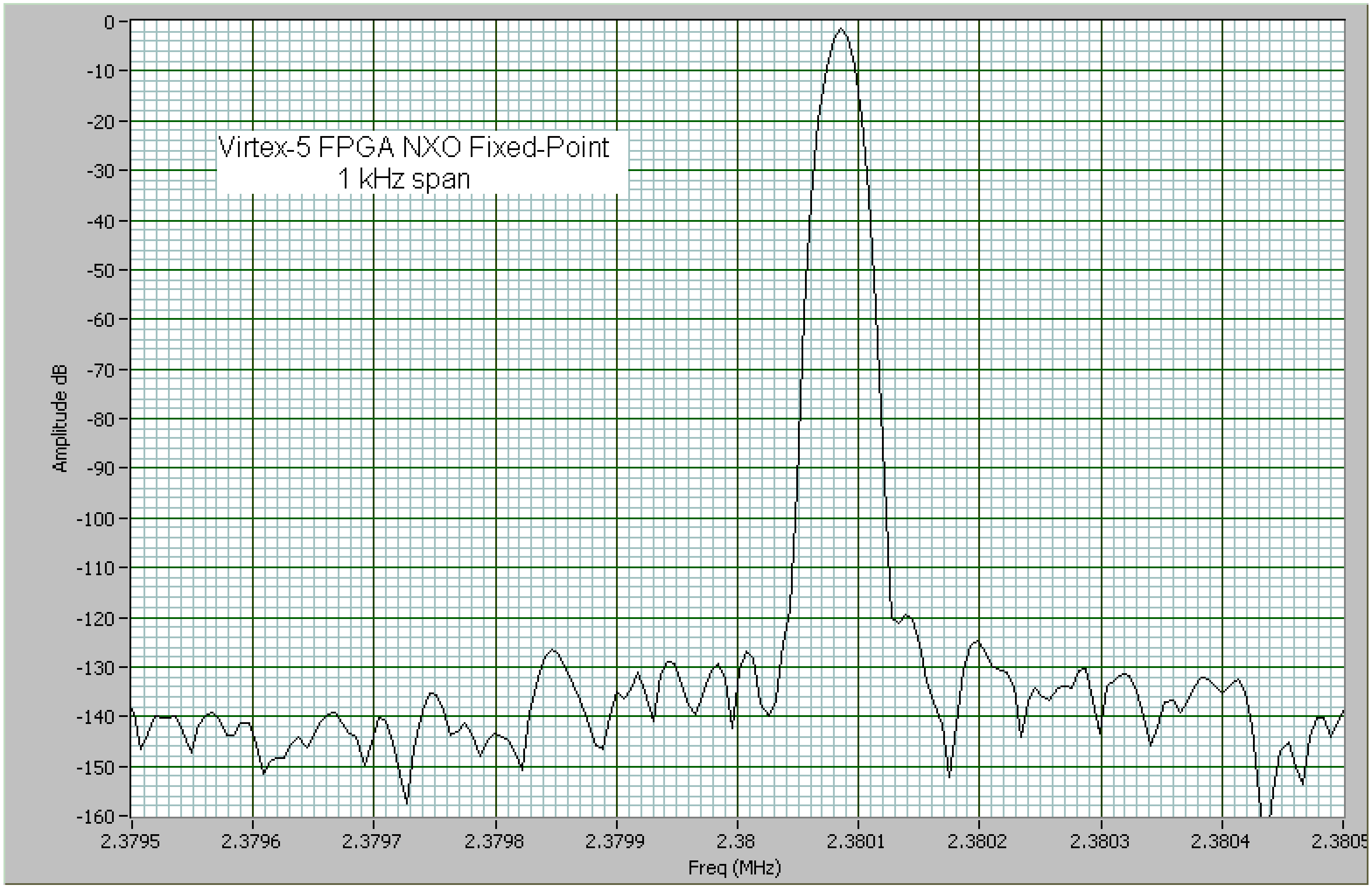
- Faster cycle time to evaluation of results
- More accessibility: parallel effort & sharing
- Optimized designs through increased collaboration
- Frees us up to make mistakes

Thank You

- ni.com/labs   keywords: SDR, RCM, Macrocell
- Filename / folder: RCM.lvproj

frank.raffaeli@ni.com

Low Close-in noise


Virtex-5 FPGA NXO Fixed-Point
1 kHz span
Amplitude dB
Freq (MHz)