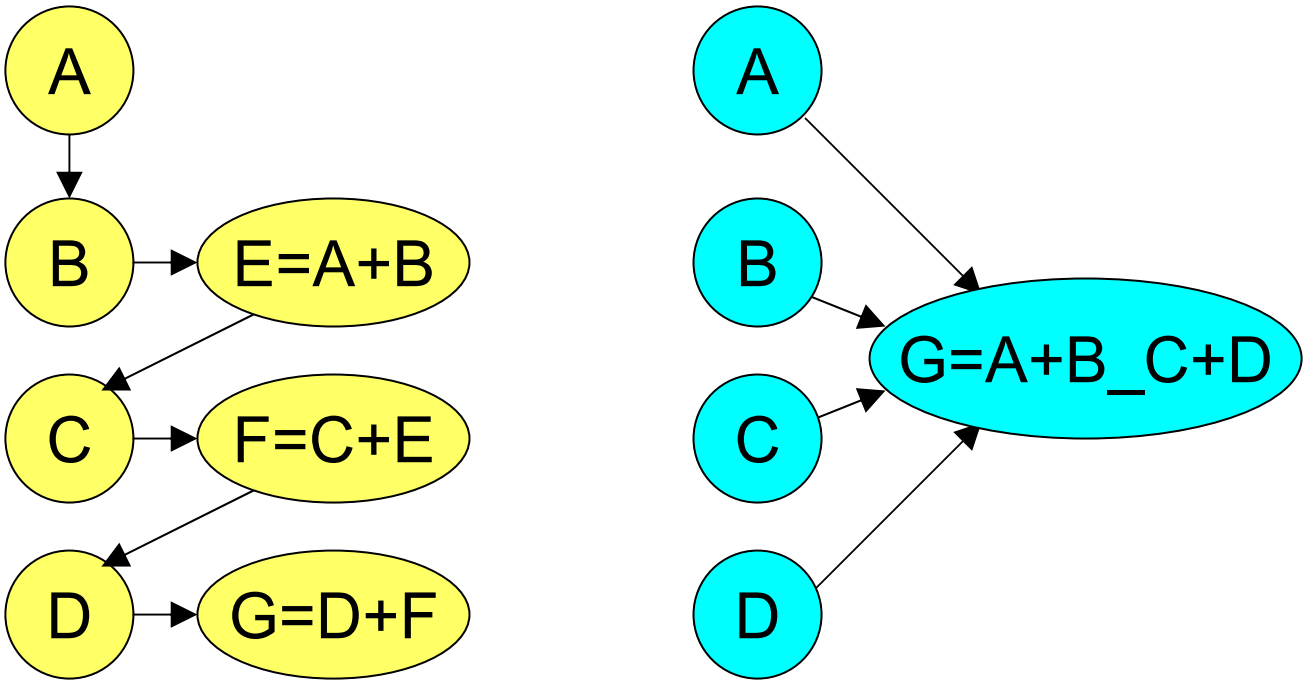
Designing for Acceleration

Ray Andraka, P.E., President
Andraka Consulting Group, Inc

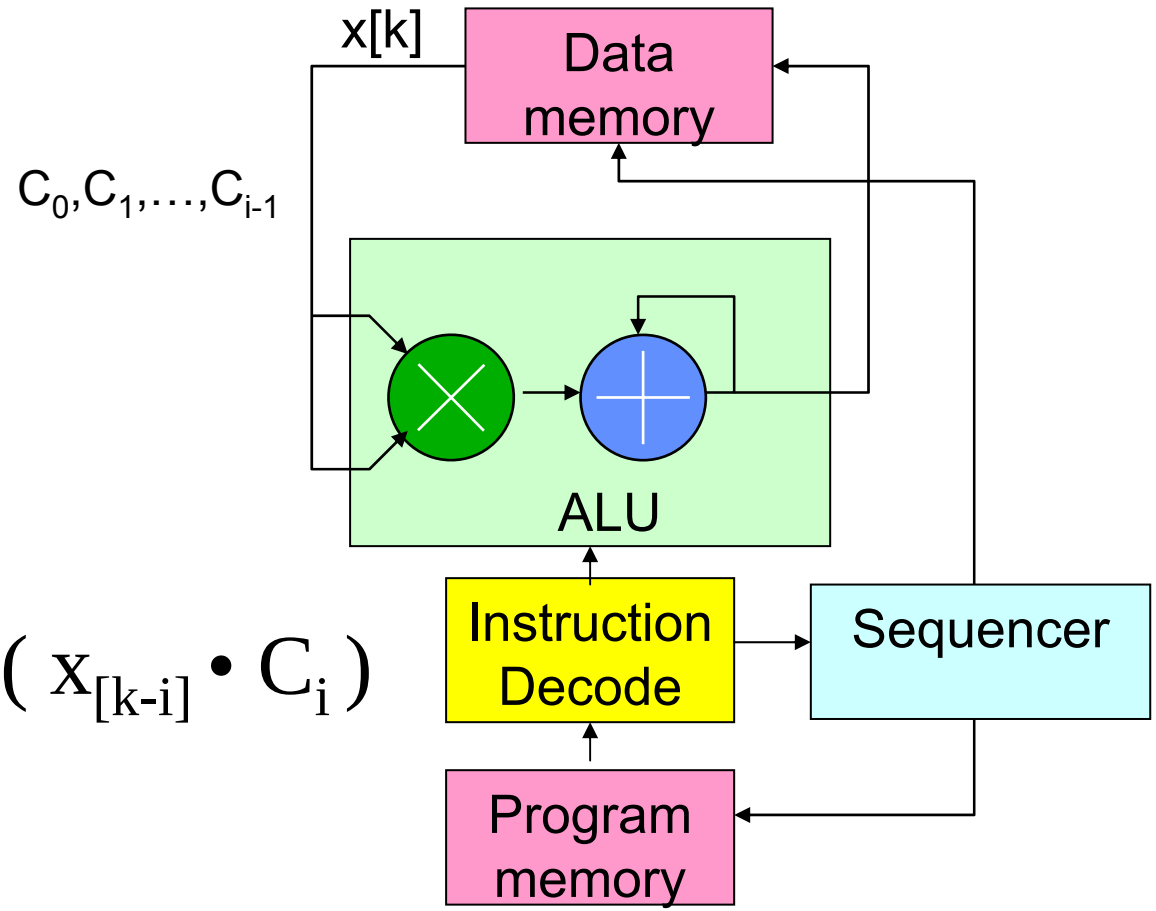
DSP with FPGAs since 1994

Why?

How?

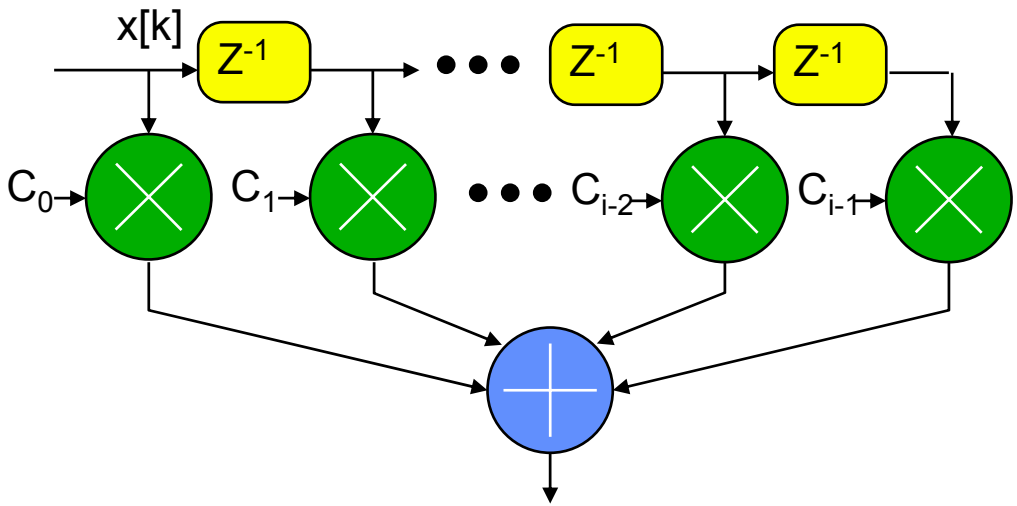


Example: Filter in Processor



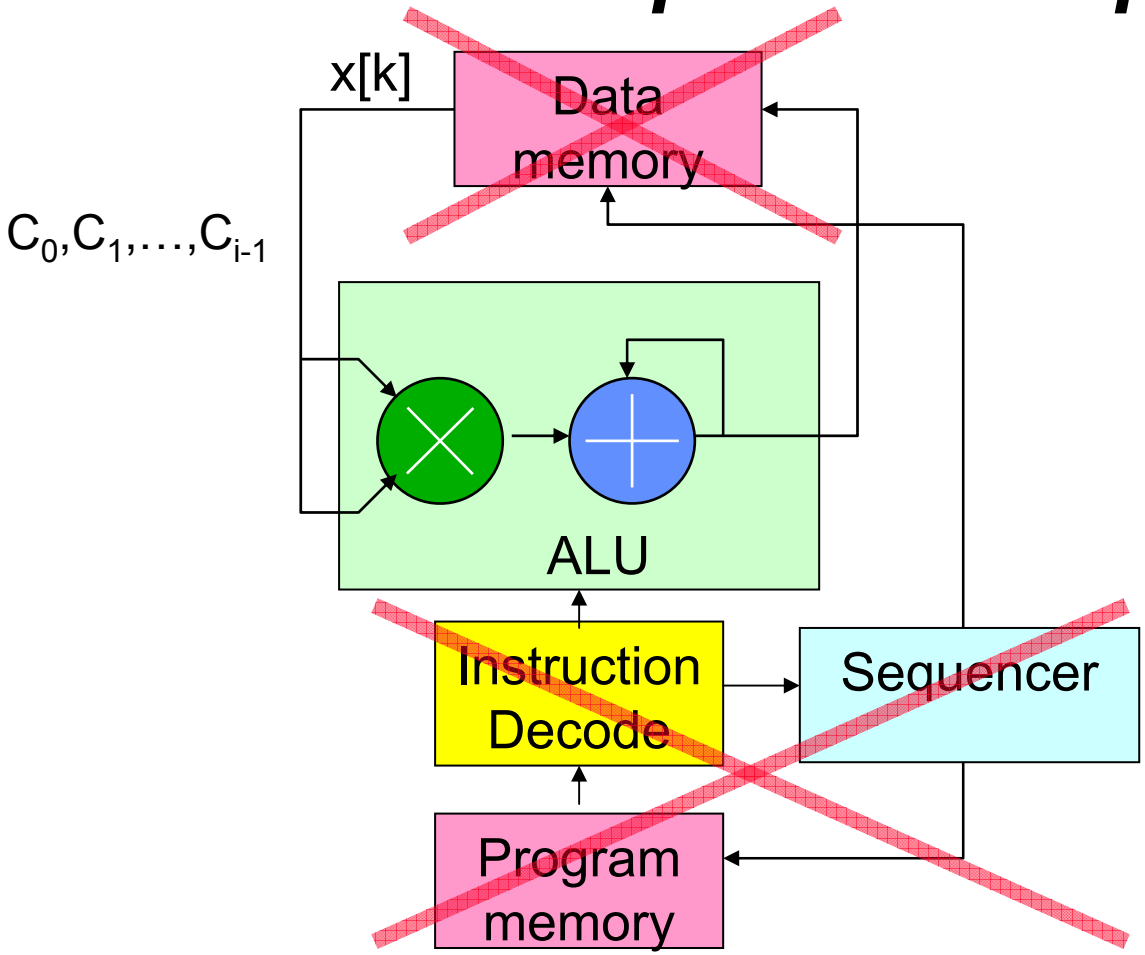
$$y_{[k]} = \Sigma_i \left(x_{[k-i]} \bullet C_i \right)$$

Example: Filter in FPGA



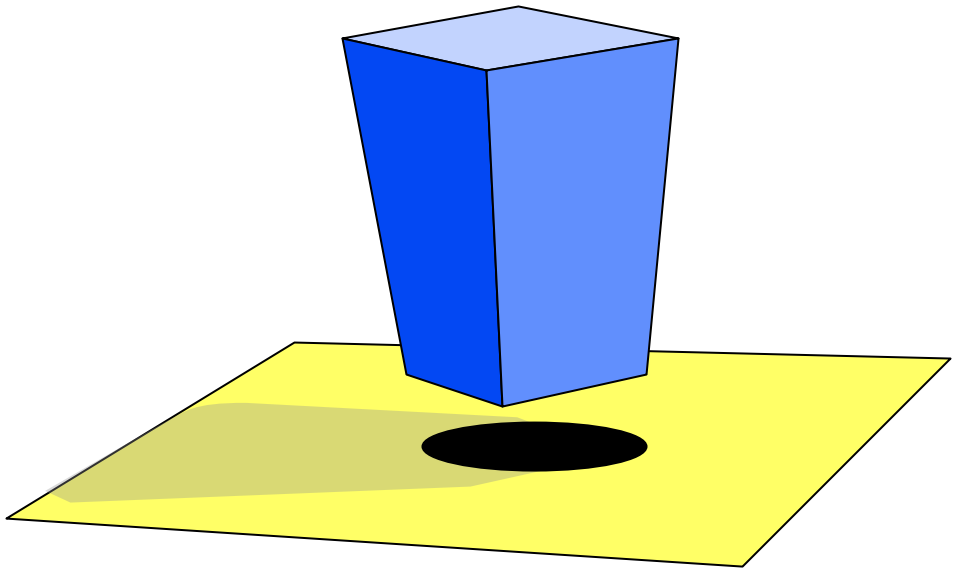
$$y_{[k]} = \Sigma_i \left(x_{[k-i]} \bullet C_i \right)$$

Lower Power *per sample*

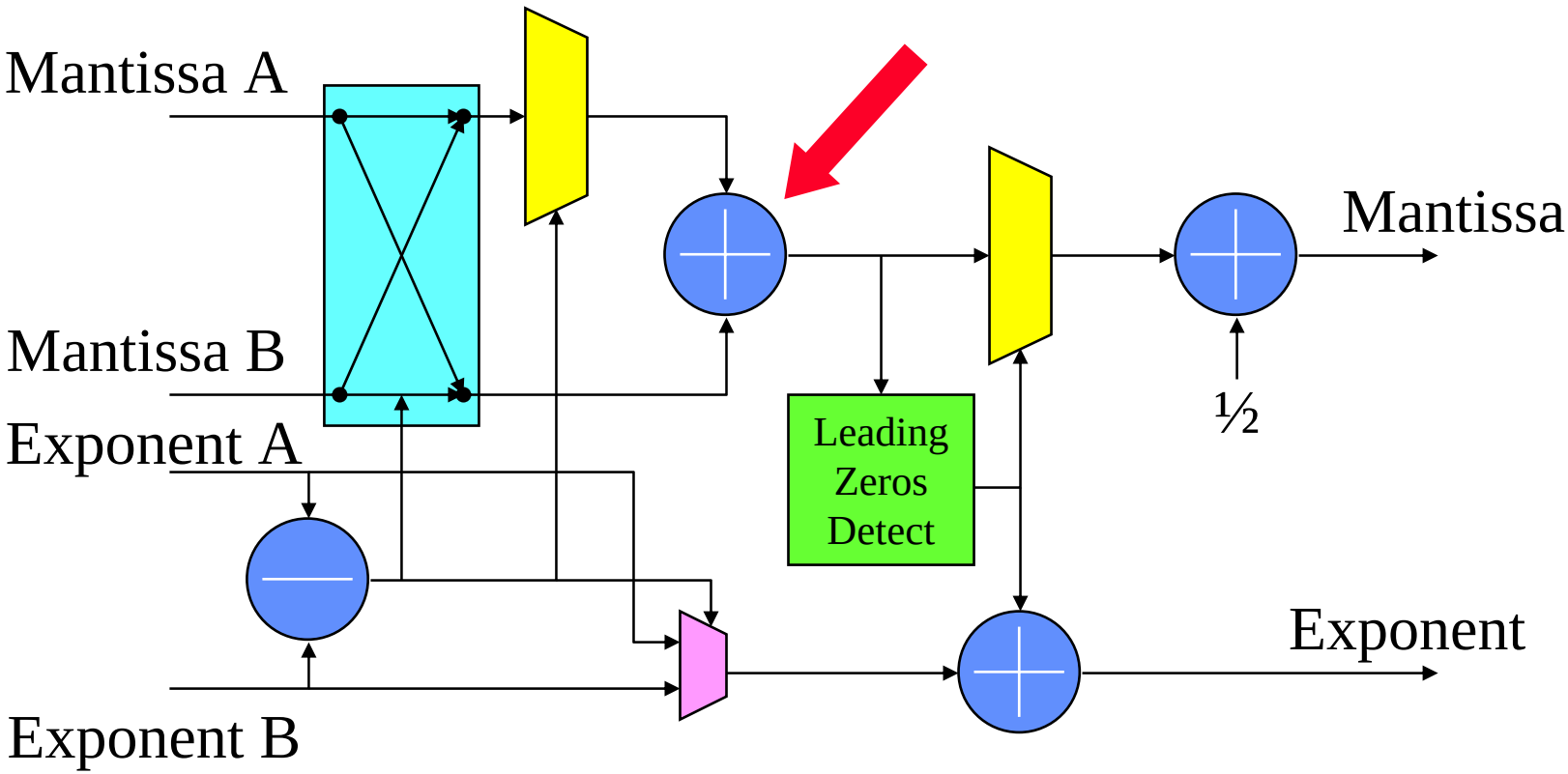


Obstacles to Success

Software Centric Algorithms

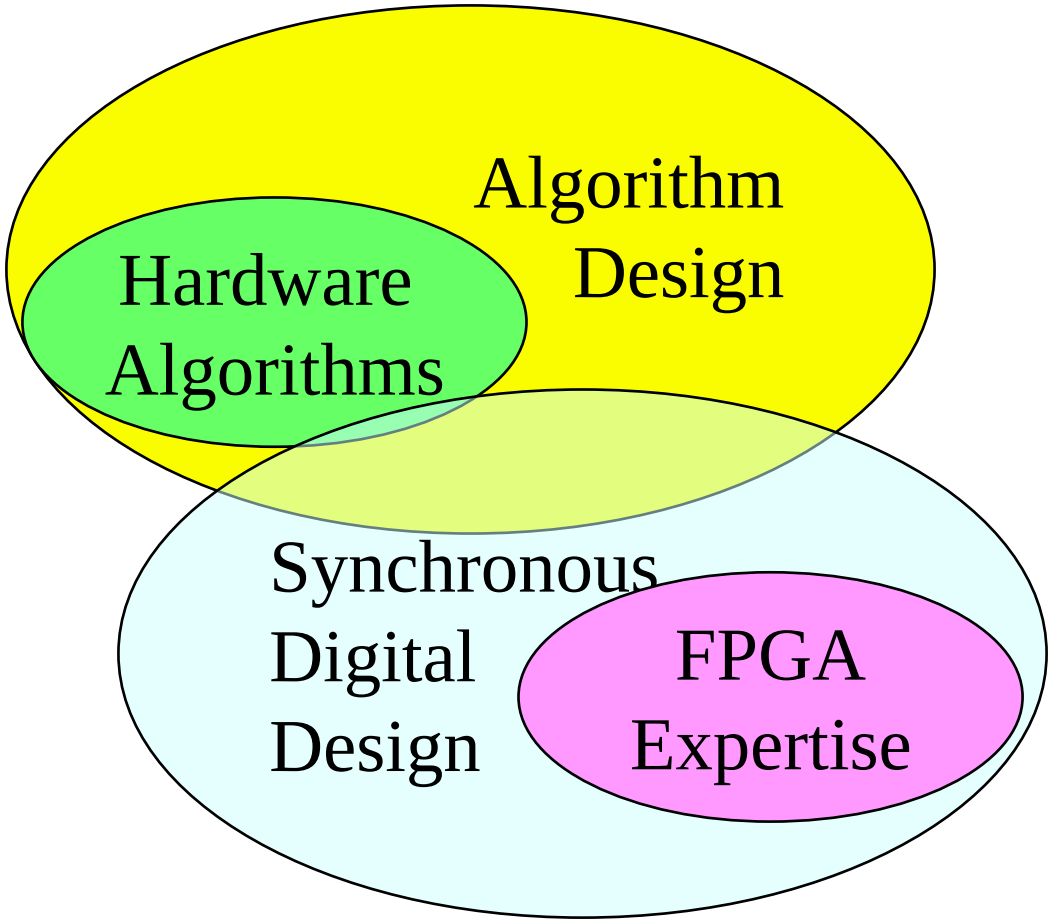


Floating Point



Immature Design Tools

Hardware DSP Expertise



Competing Design Goals

Minimum Effort

Design Space

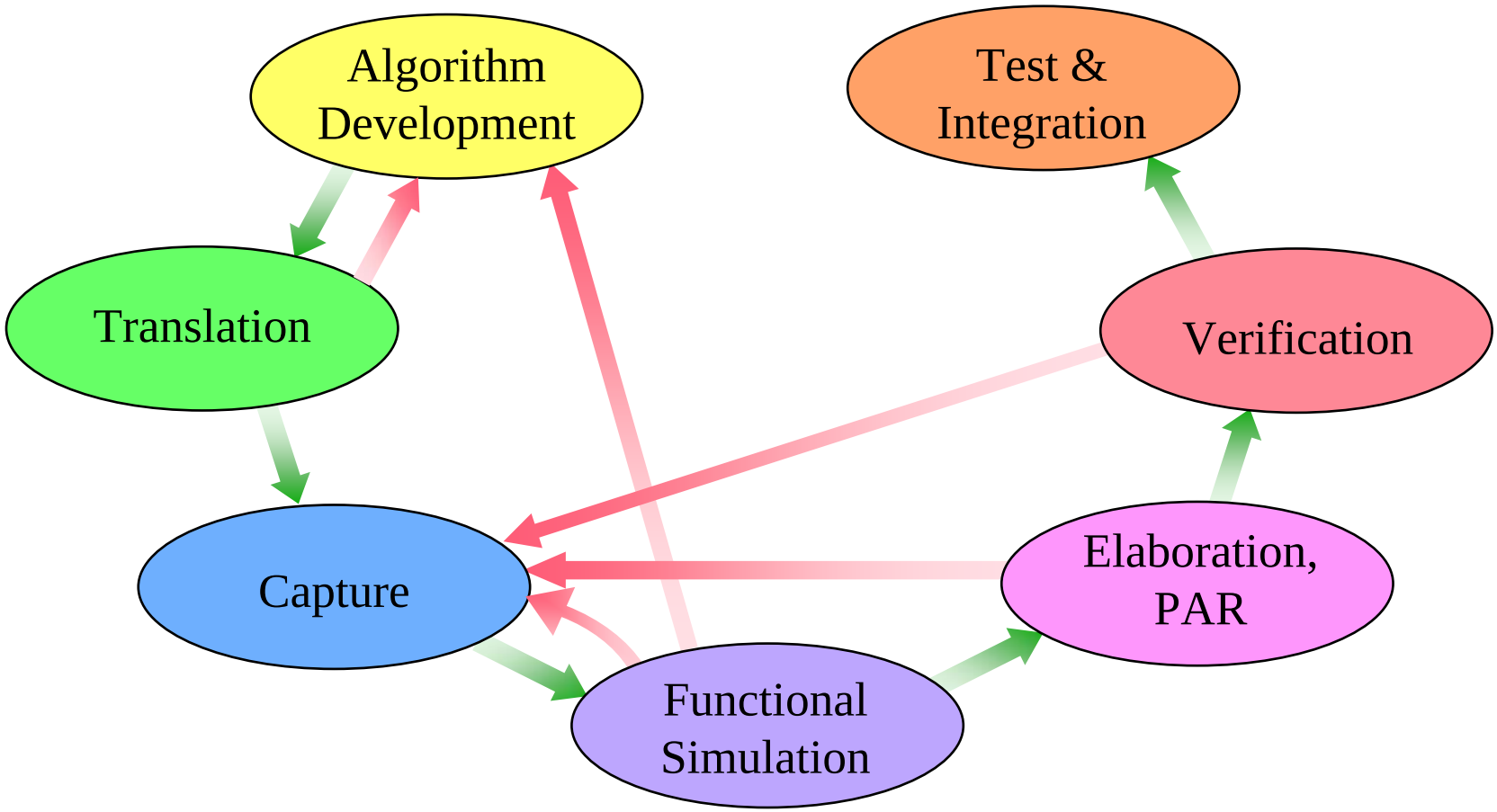
Minimum Area

Maximum Performance

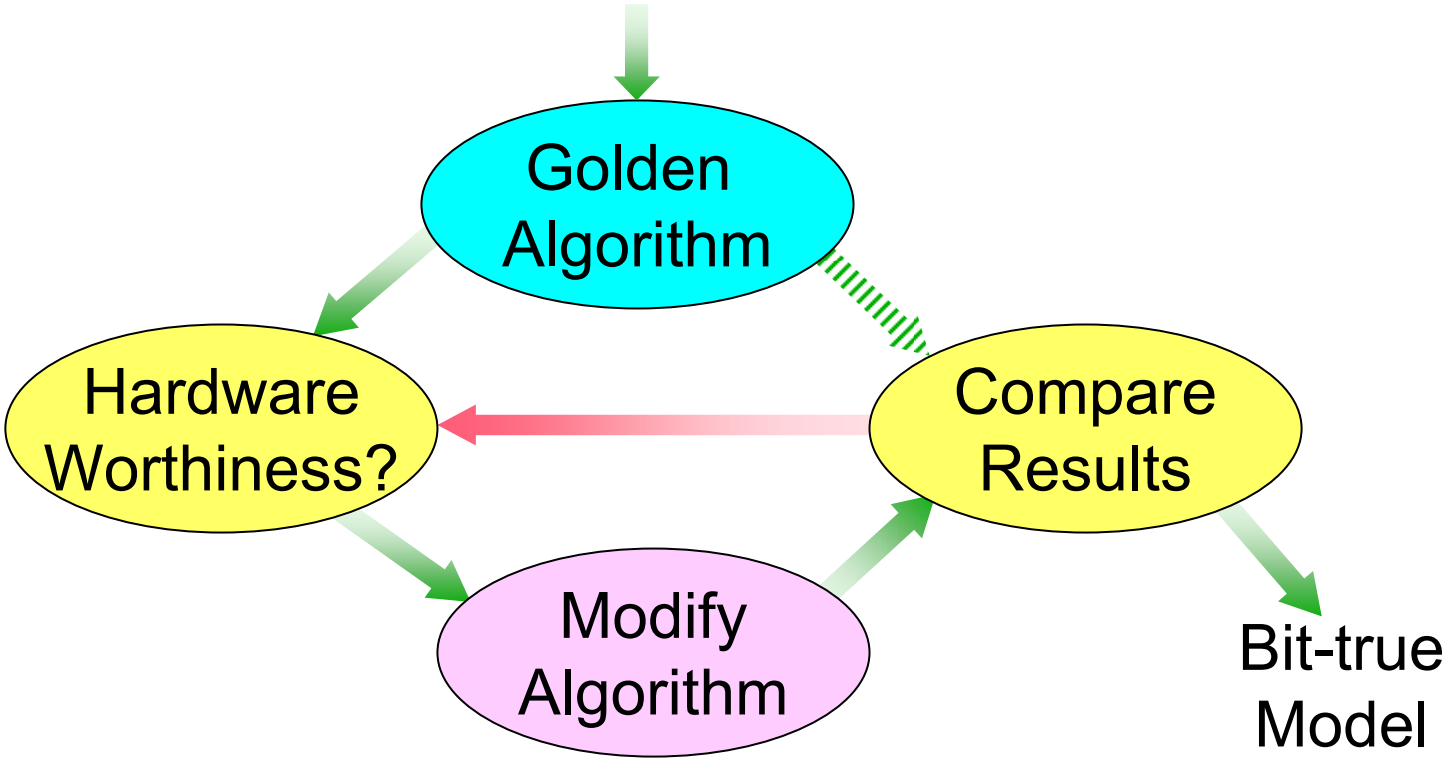
FPGA Proficiency

Tools *generate*,
they cannot *innovate*

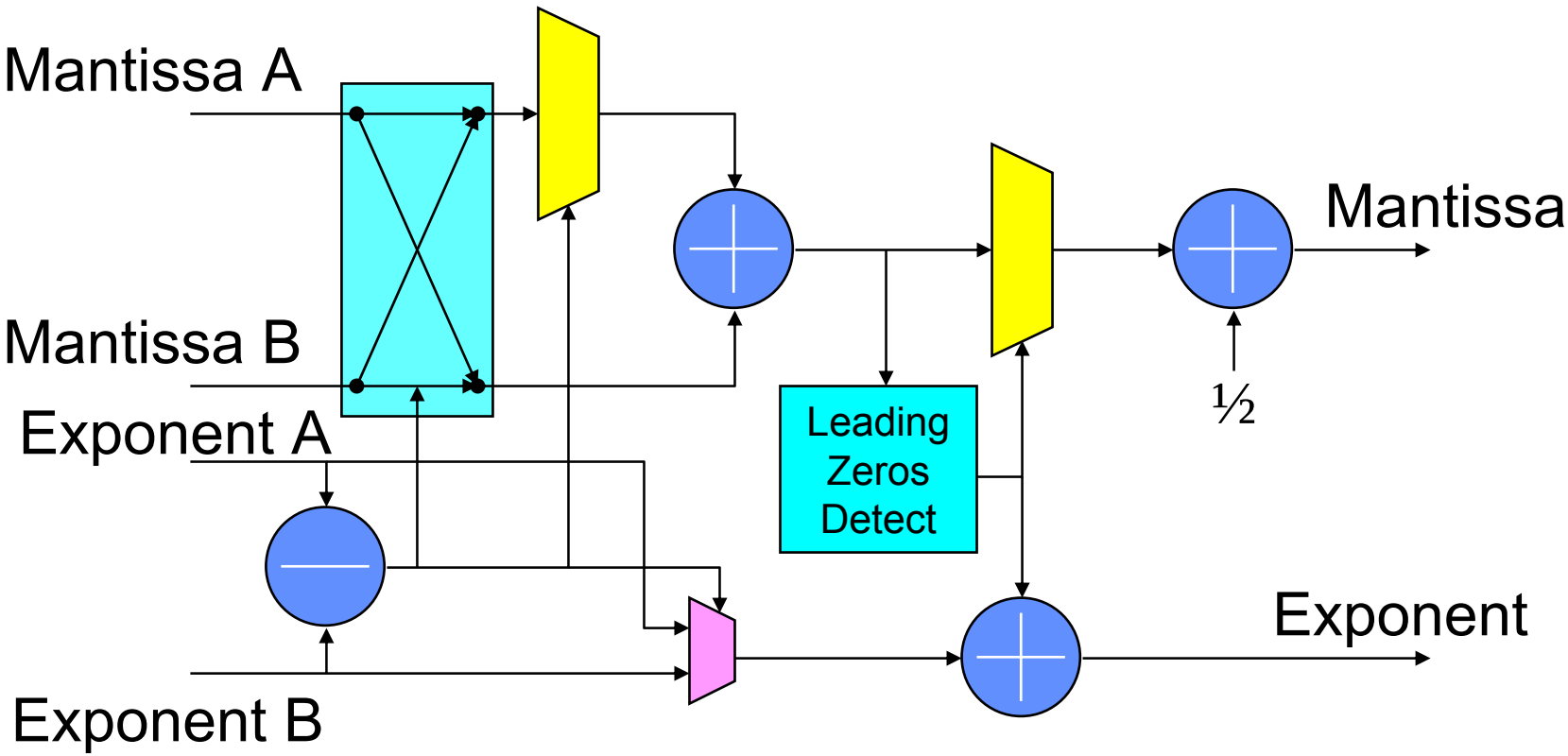
Path to Success



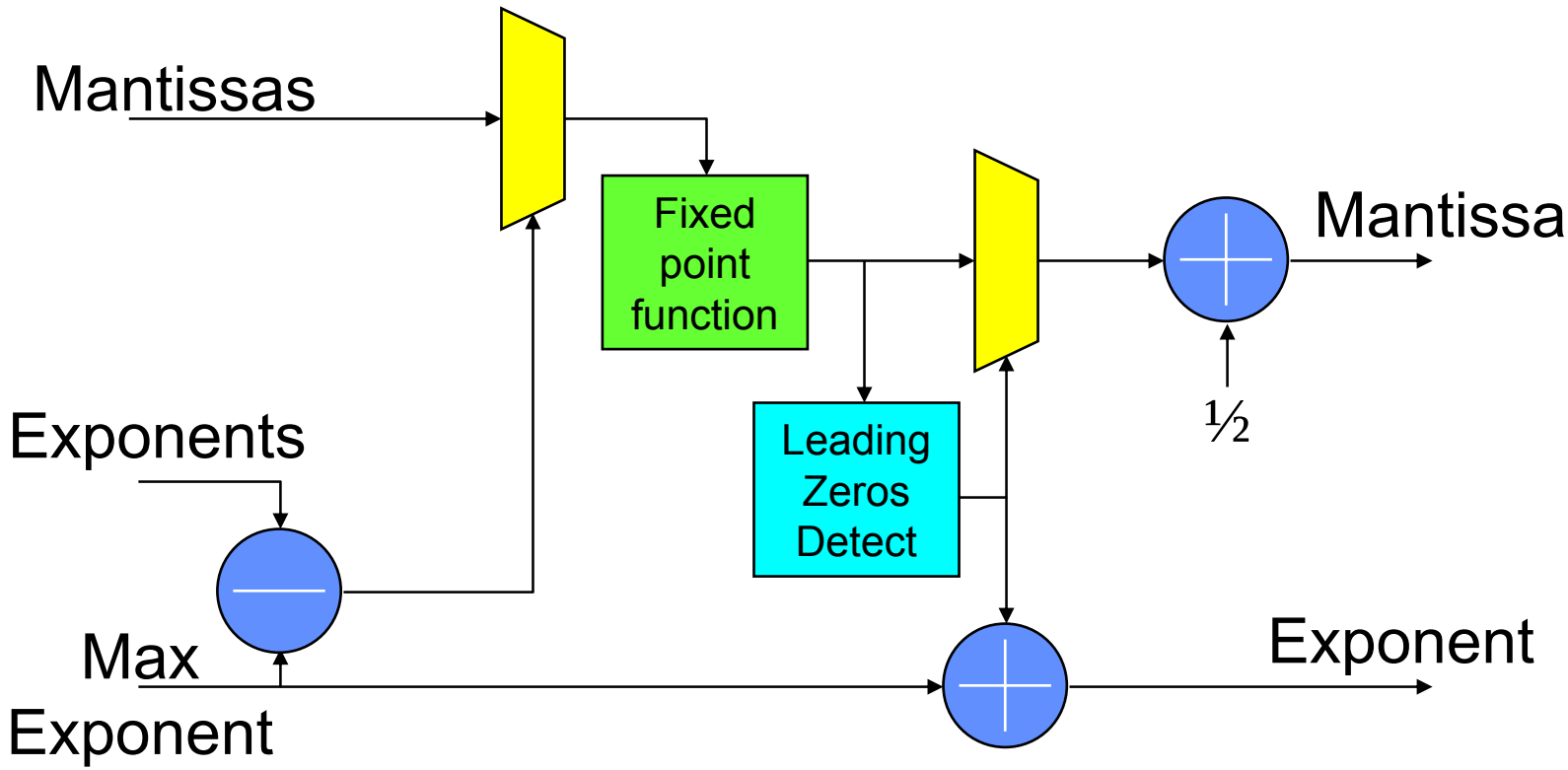
Algorithm Development



Example: Floating Point

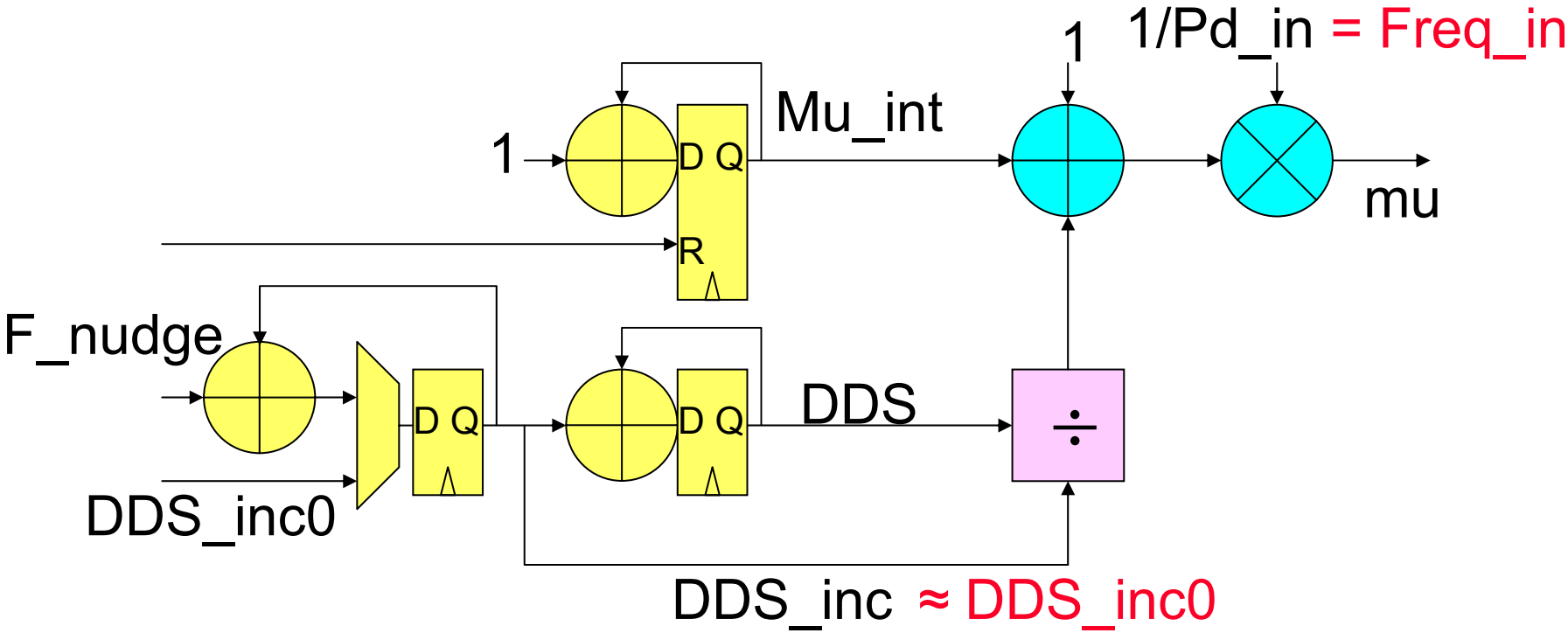


Floating Point Reduction



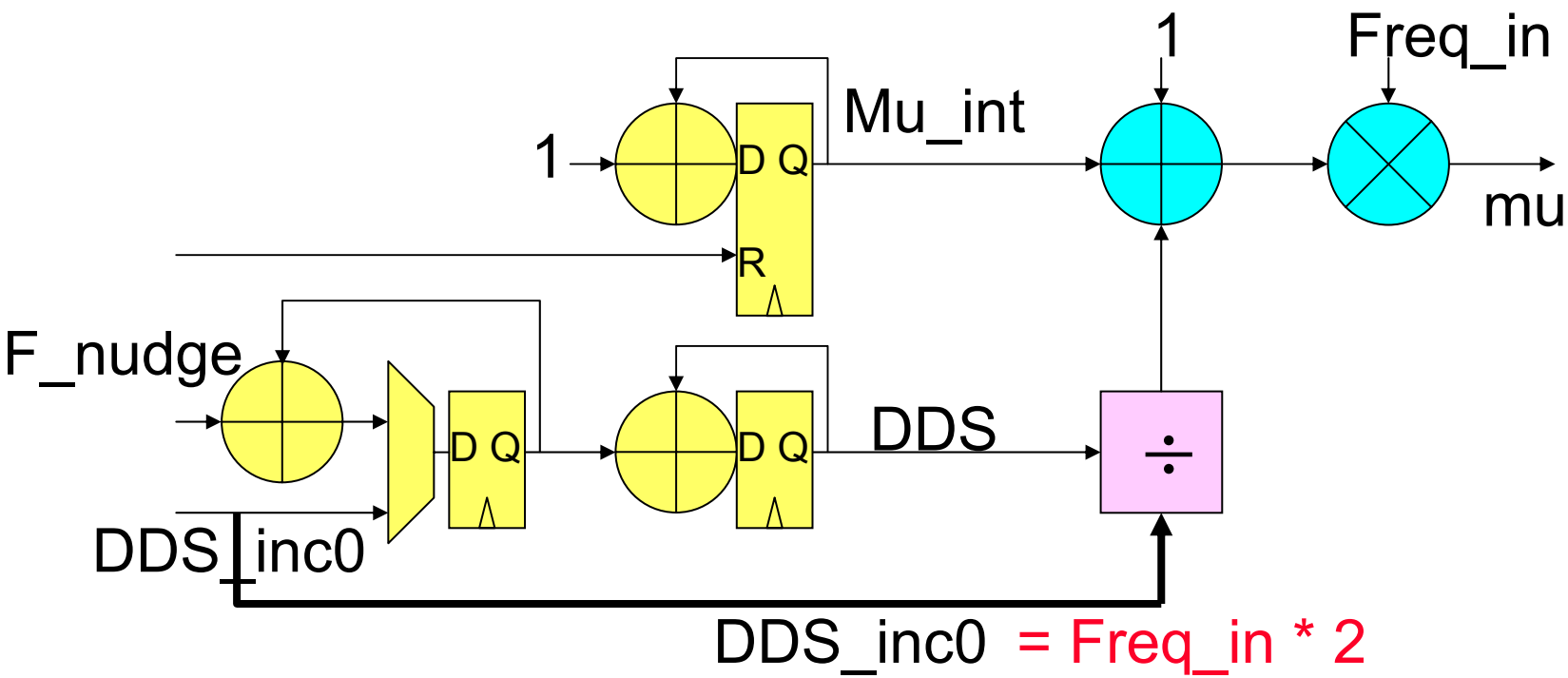
Example: Modified Algorithm

mu= (mu_int+(1-DDS/DDS_inc)) /Pd_in;



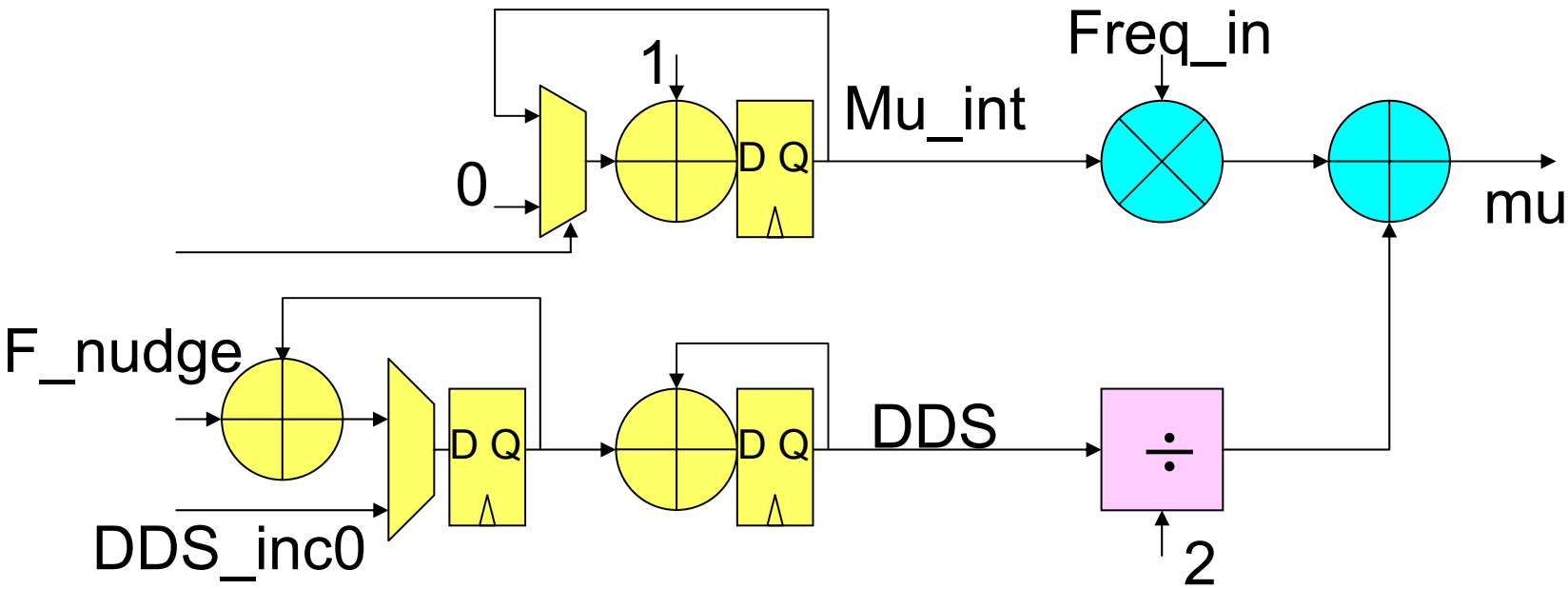
Constant Approximation

mu= (mu_int+(1-DDS/DDS_inc0)) * Freq_in;



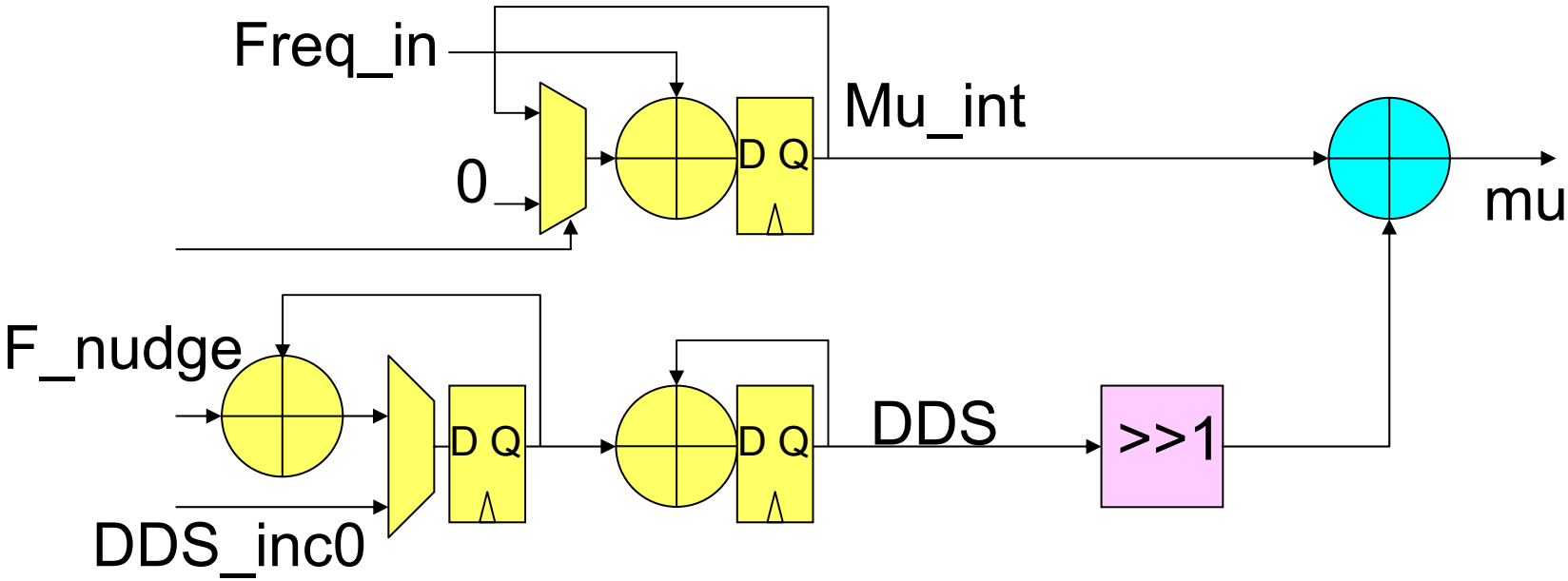
Substitution

mu= (mu_int+1) * Freq_in -DDS * ~~Freq_in~~ /(2*~~Freq_in~~);

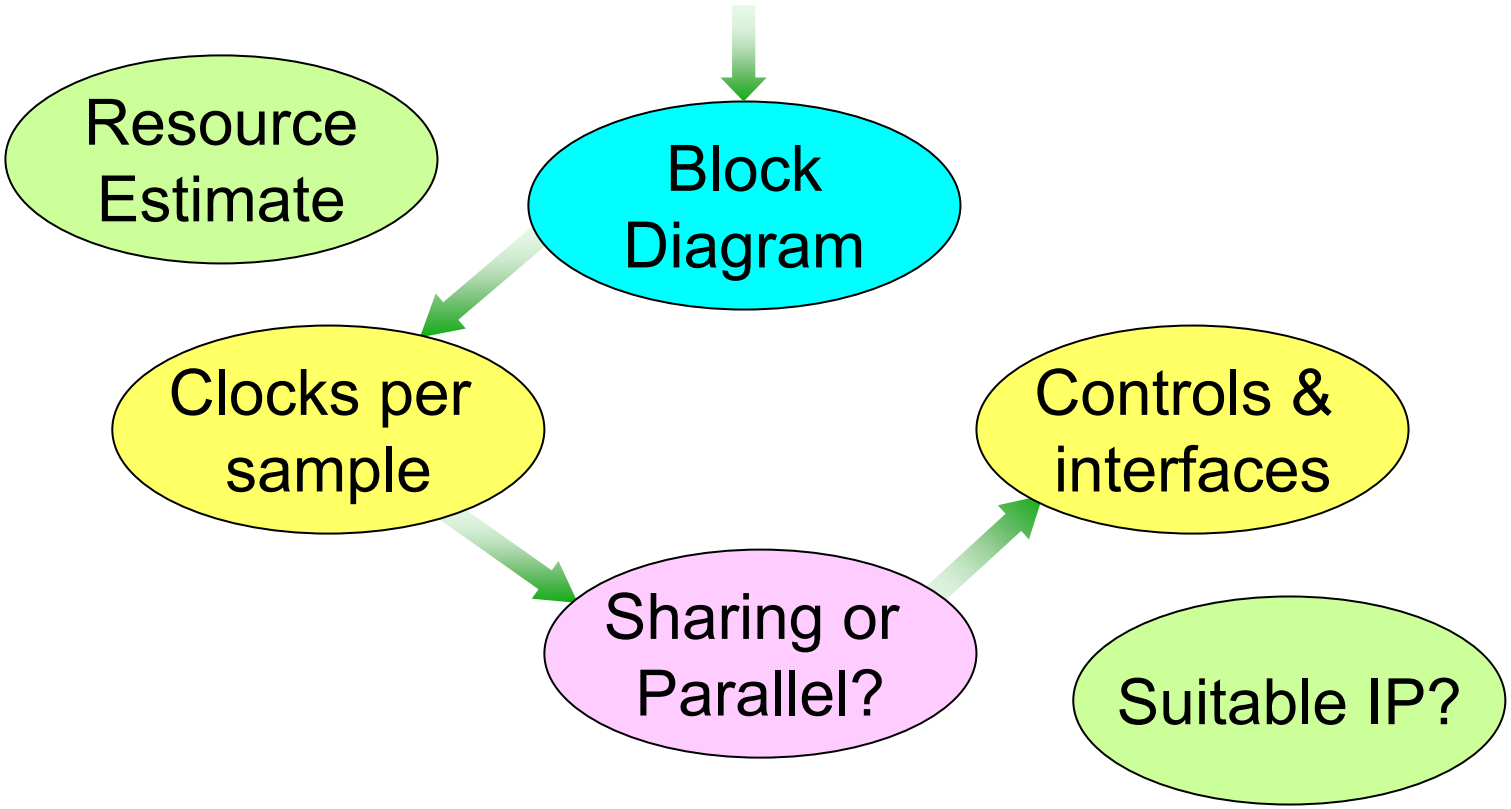


Multiply constant by count

mu= (mu_int+1) * Freq_in -DDS/2;



Translate to hardware



Block Diagram

500 MS/sec

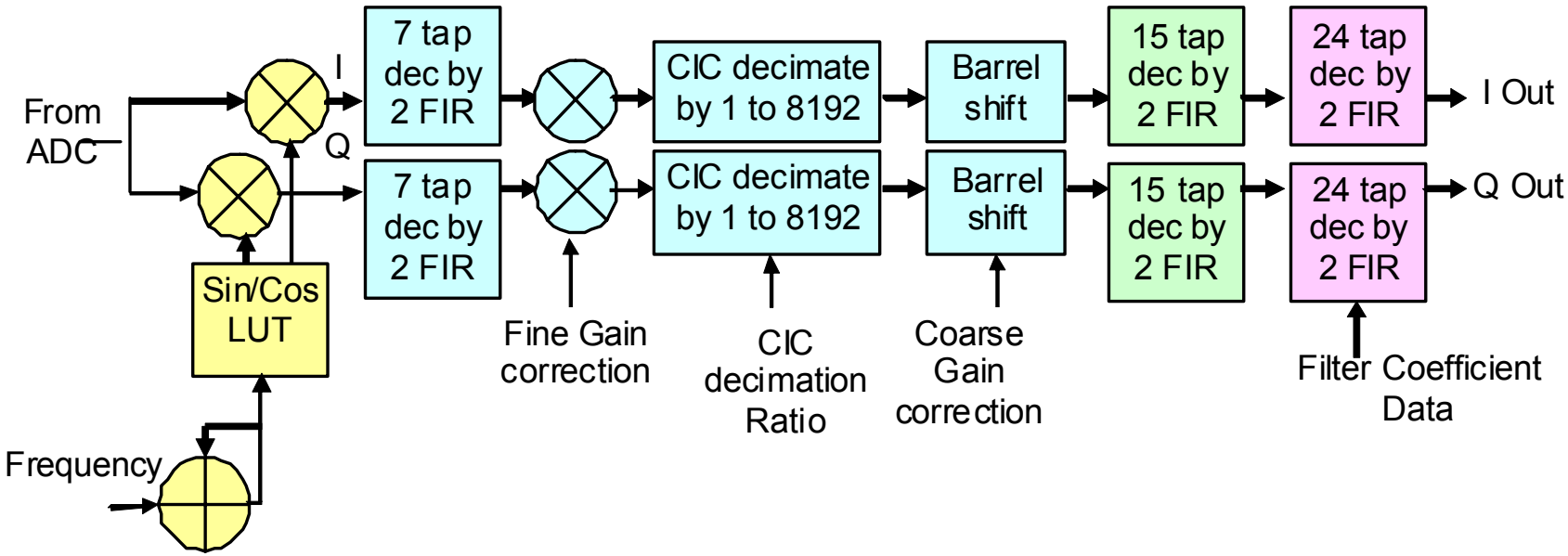
2 samples/clk

250 MS/sec

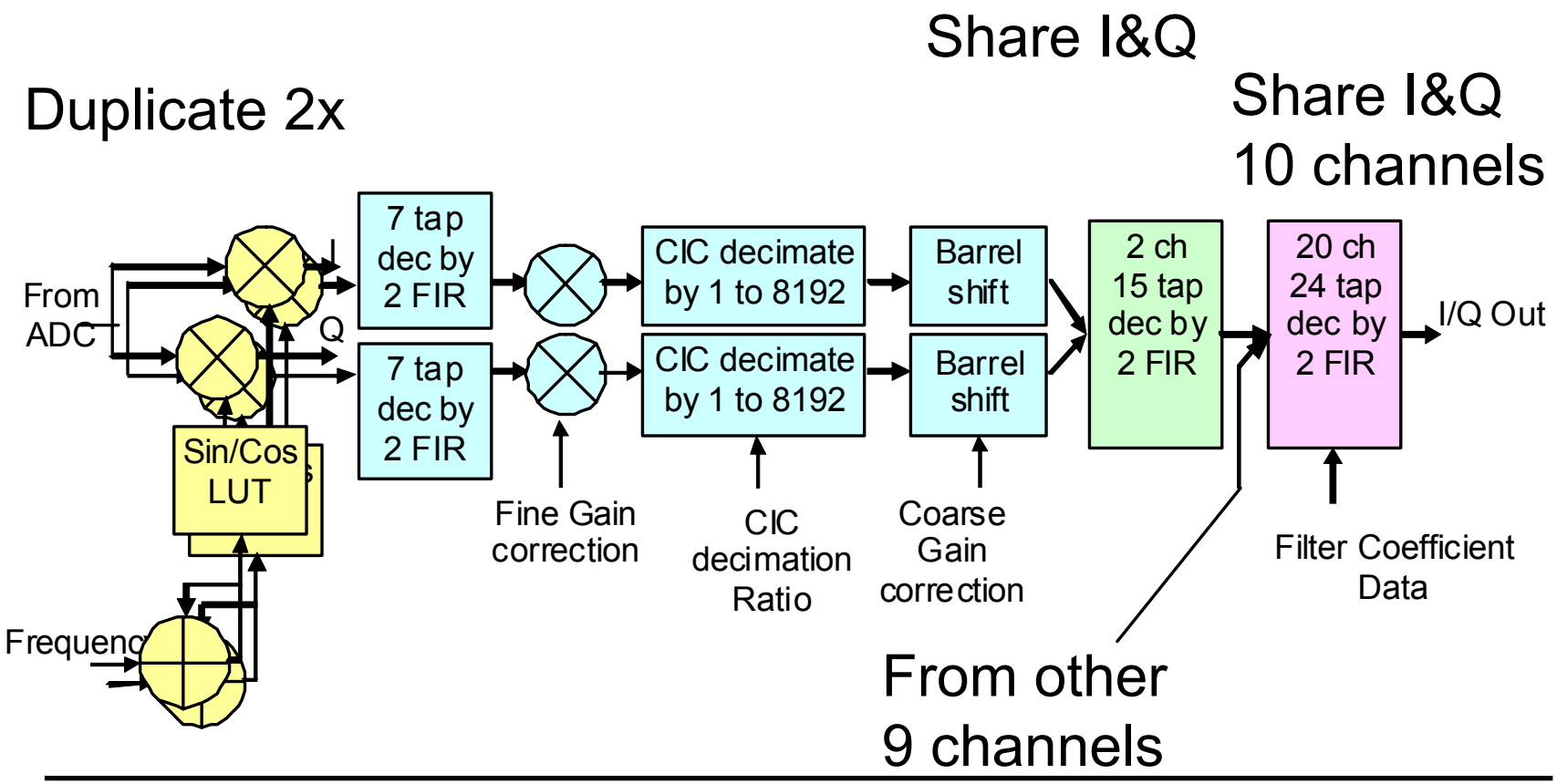
1 clk / sample

125 MS/s max

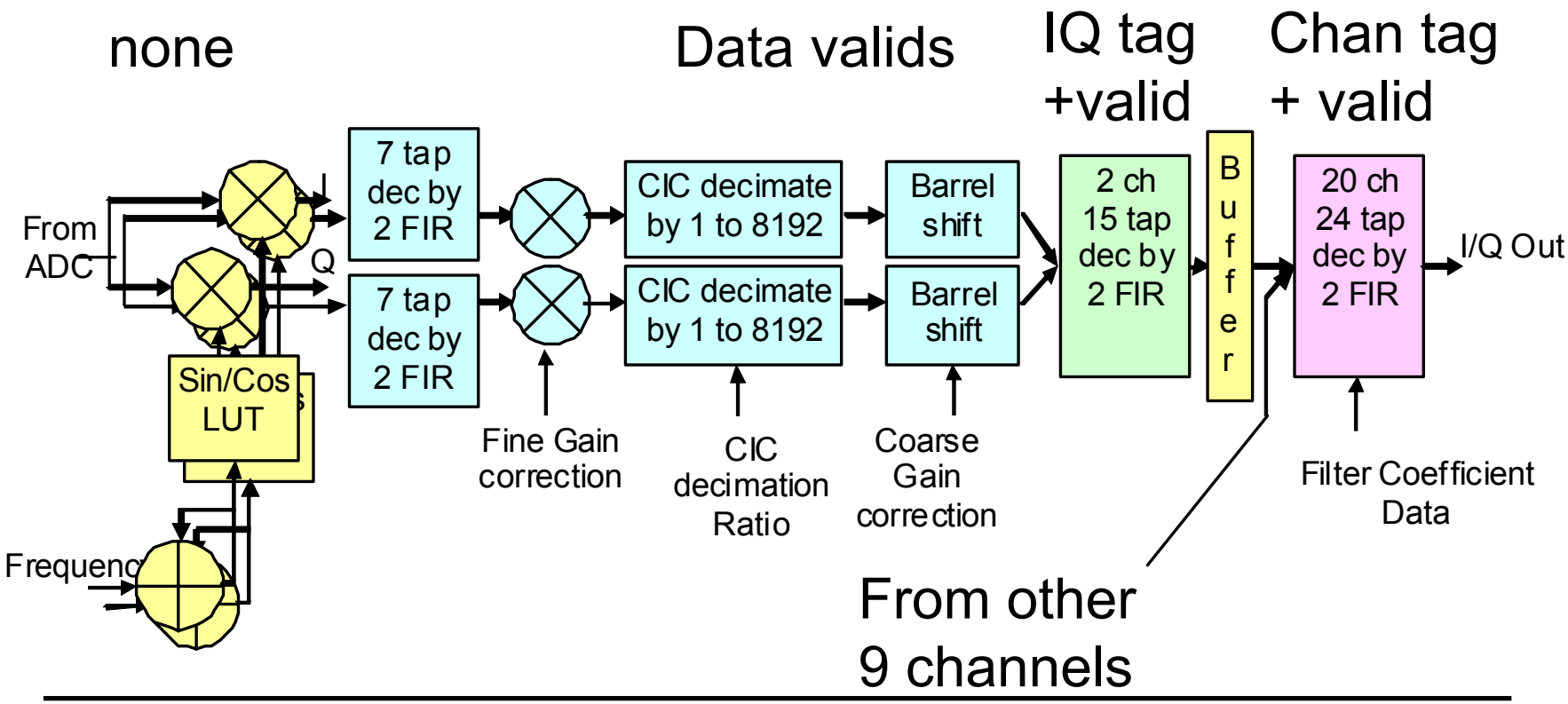
2 clk / sample



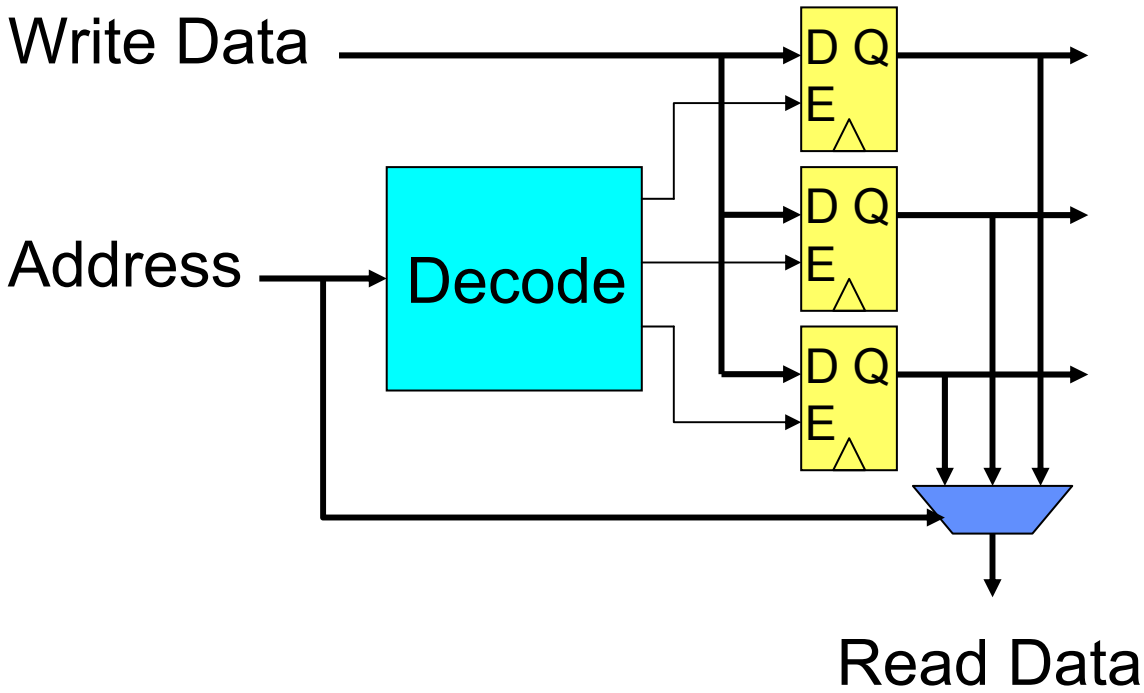
Sharing & Duplicating



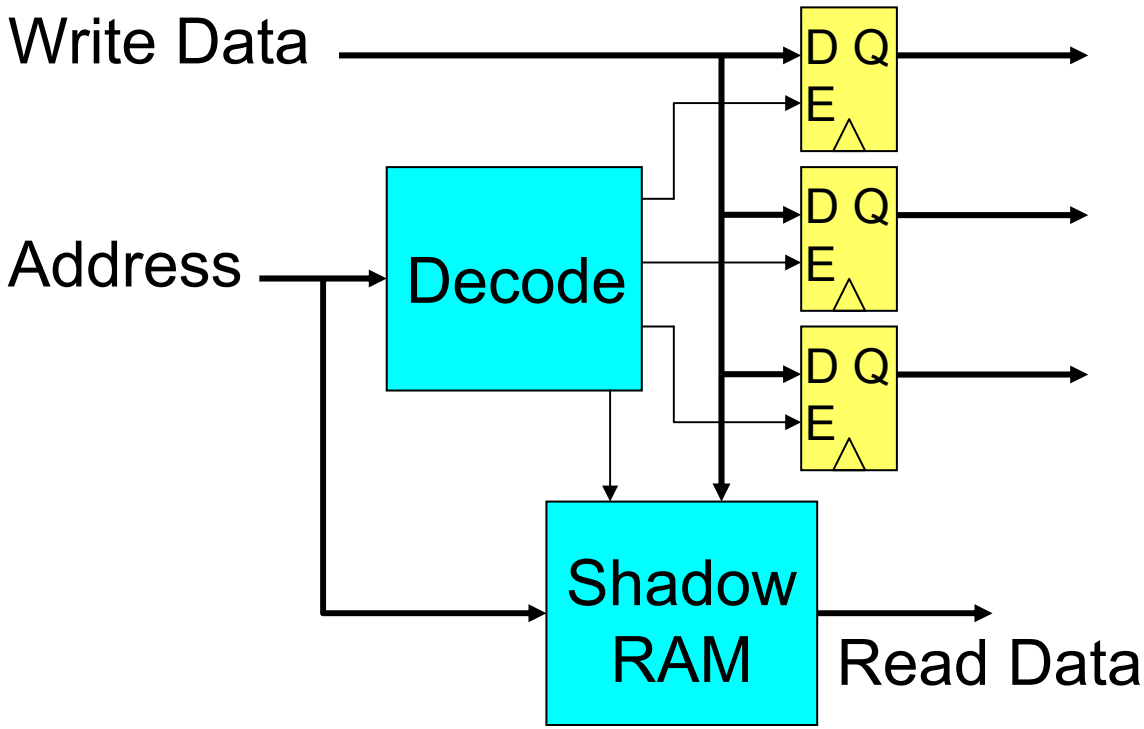
Data Flow Control



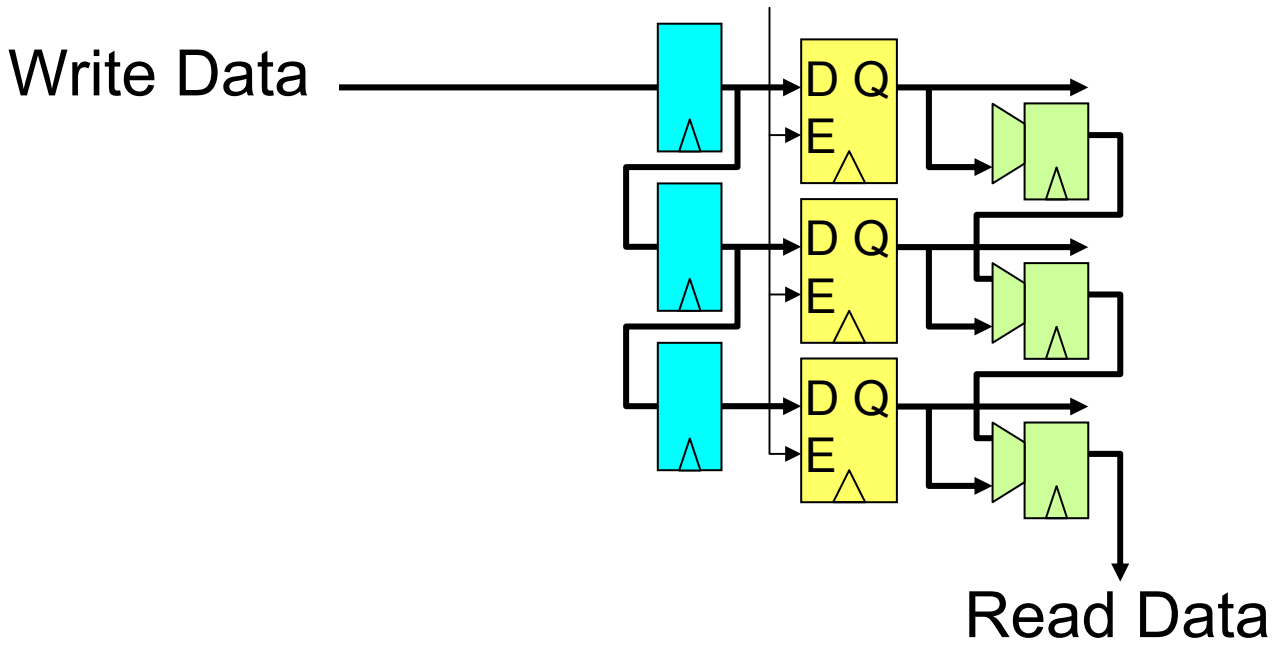
Control / Status Interface



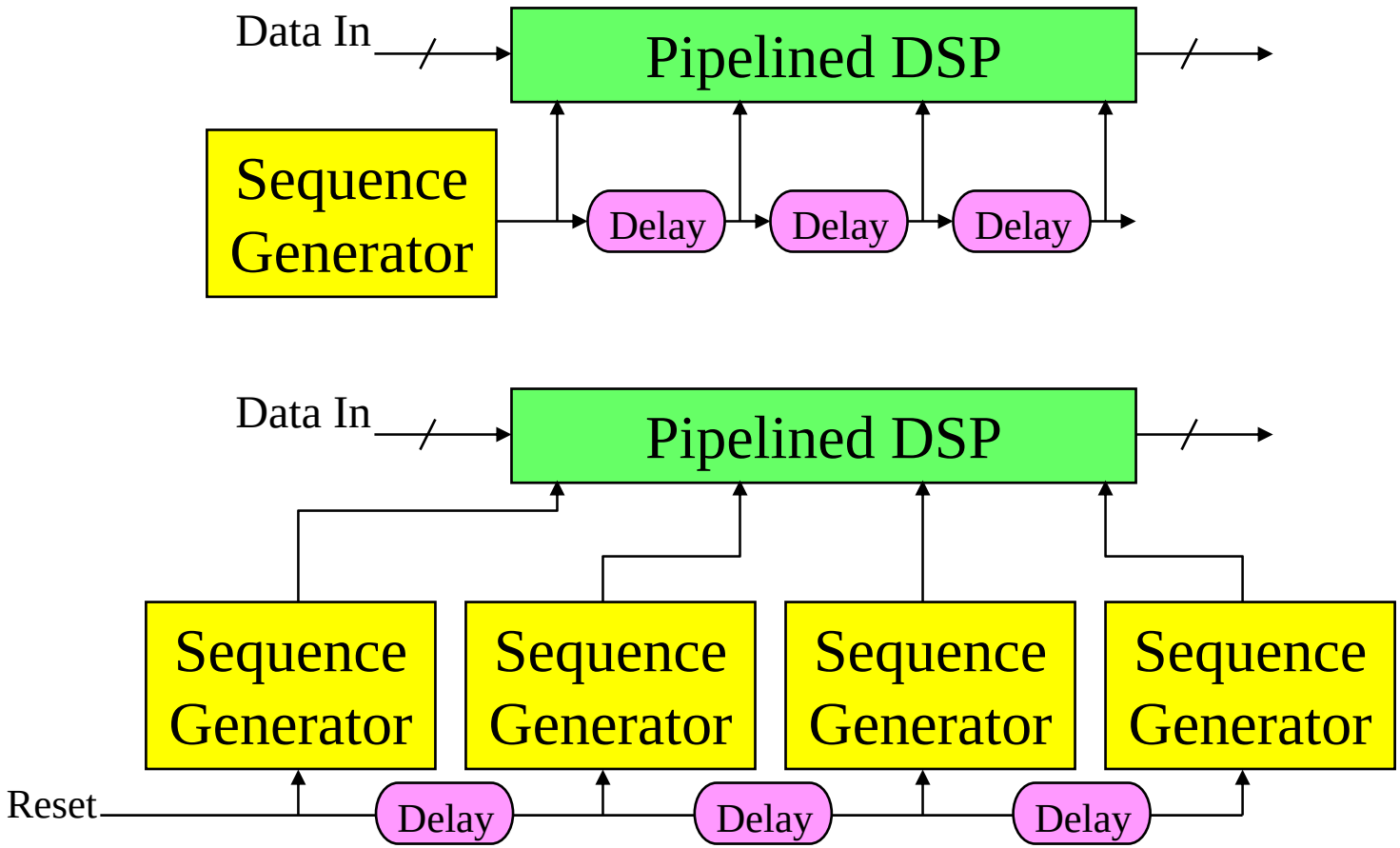
Faster Control / Status



Shift Register Access

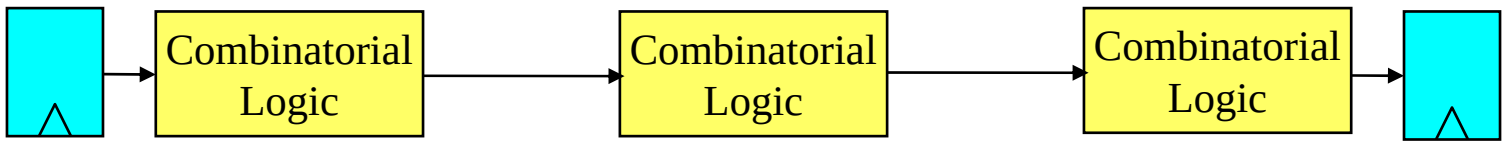


High Fan-out Controls

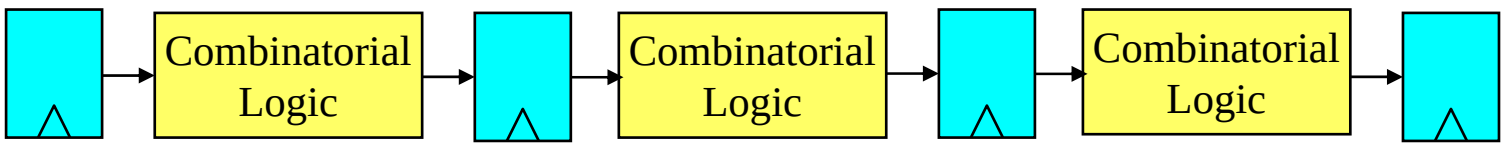


Pipelining

Minimum clock cycle = $T_{co} + \Sigma T_{pd} + T_{su} = 30ns$

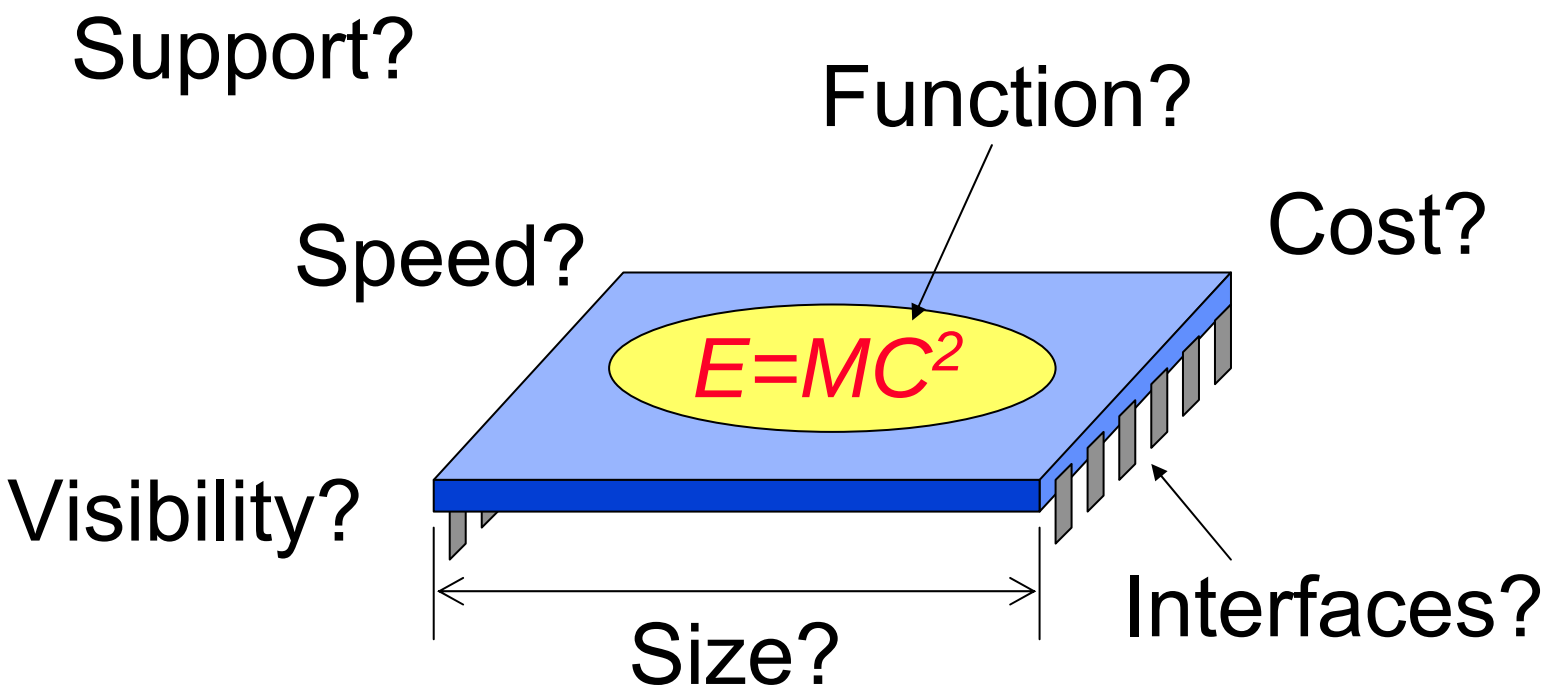


T_{co}=1ns T_{pd}=9ns T_{pd}=10ns T_{pd}=9ns T_{su}=1ns



Minimum clock cycle = $T_{co} + T_{pdmax} + T_{su} = 12ns$

Suitable IP?


Support?
Function?
Speed?
Cost?
$E=MC^2$
Visibility?
Interfaces?
Size?

Design Capture

- Which HDL?
- Hierarchical design
- Floor-planning
- Help synthesis

Simulation

- Functional only
- Bottoms up simulation
- Use hierarchy
- Test vectors from model
- Check logic correctness

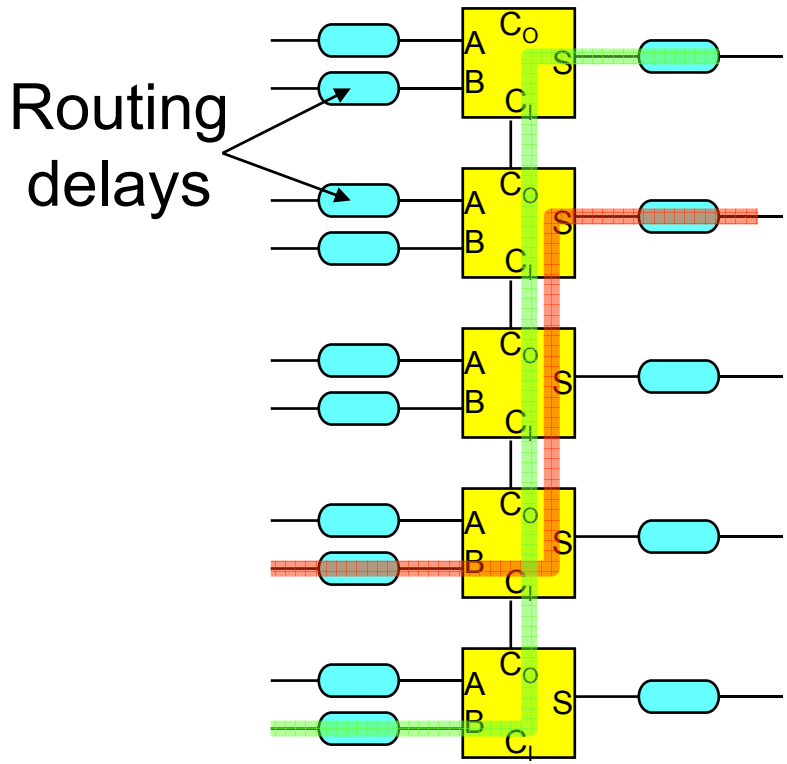
Compile

- Synthesis
- Verify structure
- Floorplanning
- Place and route

Design Verification

- Functional simulation?
- Verify pin-outs
- Static timing analysis

Timing Simulation is Bad



Test & Integration

- Use re-configurability
- Verify board first
- Simple software tests
- Share test programs

Keys to Success

- Algorithm design
- Use Hierarchy
- Exploit parallelism
- Simplify controls