OAKLAND UNIVERSITY

School of Engineering and Computer Science

Electrical and Computer Engineering

RECRLab
Reconfigurable Computing Research Laboratory

Dynamic Dual Fixed-Point CORDIC Implementation

Andres Jacoby, Daniel Llamocca

ajacobykrateil@oakland.edu, llamocca@oakland.edu

Abstract

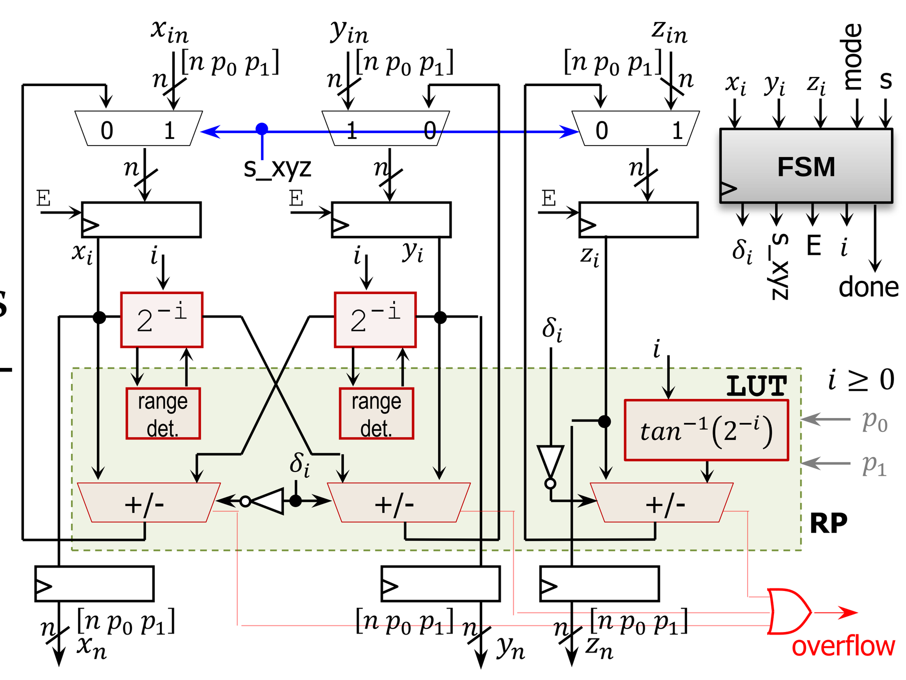
We introduce Dynamic Dual Fixed Point (DDFX) CORDIC, that provides an extra layer of flexibility to Dual Fixed-Point (DFX) CORDIC via run-time alteration of the DFX numerical format and features enhanced dynamic range and accuracy.

DDFX CORDIC is compared with Dual Fixed Point, Fixed Point, and Floating Point realizations in terms of resources and accuracy. Results show that this hardware/software approach achieves higher dynamic range than DFX CORDIC at the expense of a slight resource increase, while exhibiting comparable of even higher accuracy than Floating Point CORDIC realizations.

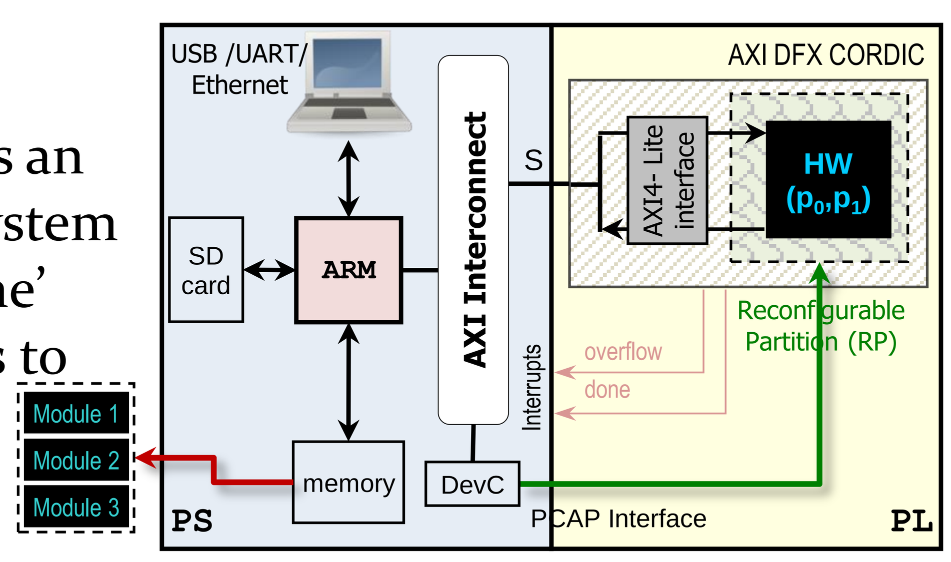
DDFX CORDIC implementation

- DDFX format: set of DFX formats sharing the same wordlength.
- Example: [32 26 4 6]: DFX formats [32 26 20], [32 18 12], [32 10 4].
- DFX format: [n p_0 p_1]: Acts as having 2 FX formats [n-1 p_0], [n-1 p_1].
- DDFX is implemented by modifying the DFX format at run-time.

DDFX CORDIC: We use the DFX CORDIC hardware. We adjust the DFX format (vary p_0 and p_1 with n fixed) by altering the combinational logic that depends on p_0 and p_1; this is the run-time alterable area (green-shaded). 'overflow': it signals a need for a numerical format with higher dynamic range. The DFX format is to be adjusted at run-time based on arithmetic overflow and user input.



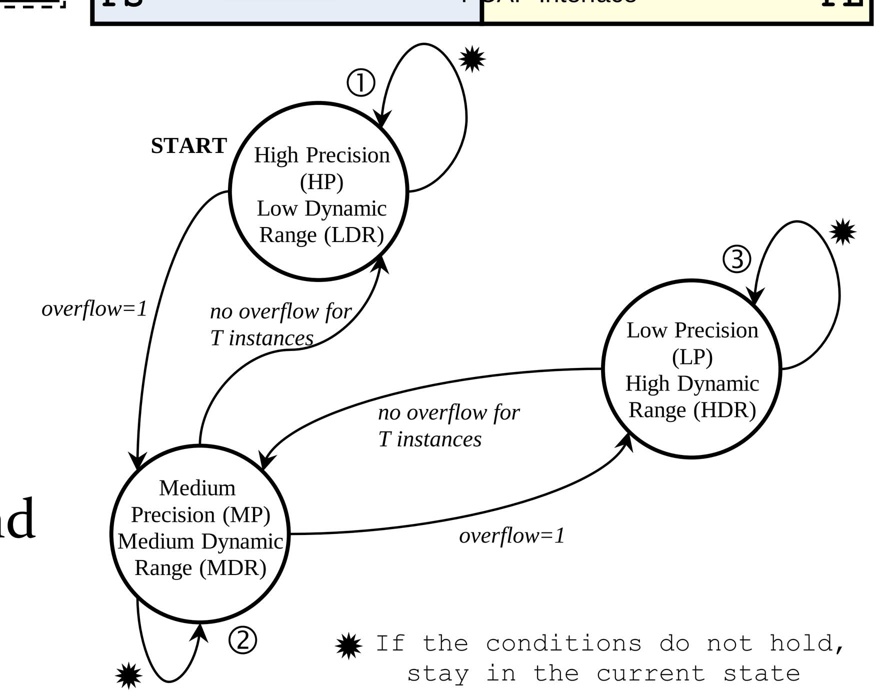
Self-reconfigurable embedded system: DFX CORDIC is placed as an AXI peripheral in an embedded system (ZYBO Board). 'overflow' and 'done' are connected to interrupt sources to trigger run-time reconfiguration.



Run-Time management

DDFX CORDIC is implemented by dynamically swapping DFX CORDIC cores to meet requirements based on overflow or user input.

The behavior is specified as a State Diagram where the transitions depend on overflow and requests by user:



State Diagram: Implemented by a software routine.

- Each state represents a hardware configuration in a certain DFX format, providing a trade-off of dynamic range and precision.
- If overflow is asserted: we trigger reconfiguration to a higher dynamic range/lower precision configuration.
- If we have T overflow-free computations, we trigger reconfiguration to a lower dynamic range/high precision configuration.

Experimental Setup

DDFX formats tested (32 and 24 bits), along with the DFX, FX, FP CORDIC realizations we perform comparisons against:

DDFX Format	Set of DFX formats	Range of Values	Max/Min Resolution	Dynamic Range	DFX	FX	FP
[32 26 4 6]	[32 26 20] [32 18 12] [32 10 4]	$[-2^{26}, 2^{26}-2^{-4}]$	1.49×10^{-8} 6.25×10^{-2}	313.07dB	[32 10 4]	[32 4]	32-bit
[32 29 1 8]	[32 29 21] [32 19 11] [32 9 1]	$[-2^{29}, 2^{29}-2^{-1}]$	1.86×10^{-9} 5×10^{-1}	349.12dB	[32 9 1]	[32 1]	32-bit
[24 20 4 4]	[24 20 16] [24 14 10] [24 8 4]	$[-2^{18}, 2^{18}-2^{-4}]$	9.54×10^{-7} 6.25×10^{-2}	228.78dB	[24 8 4]	[24 4]	24-bit EW:7, FW:16
[24 23 1 6]	[24 23 17] [24 15 9] [24 7 1]	$[-2^{21}, 2^{21}-2^{-1}]$	1.19×10^{-7} 5×10^{-1}	264.91dB	[24 7 1]	[24 1]	24-bit EW:7, FW:16

Functions tested: atan(x), $A_n\sqrt{x^2+y^2}$. State Diagram: T=50.

Tested on: Xilinx XC7Z010 Zynq-7000 PSoC.

Results

Resources (Xilinx 7-Series Slices):

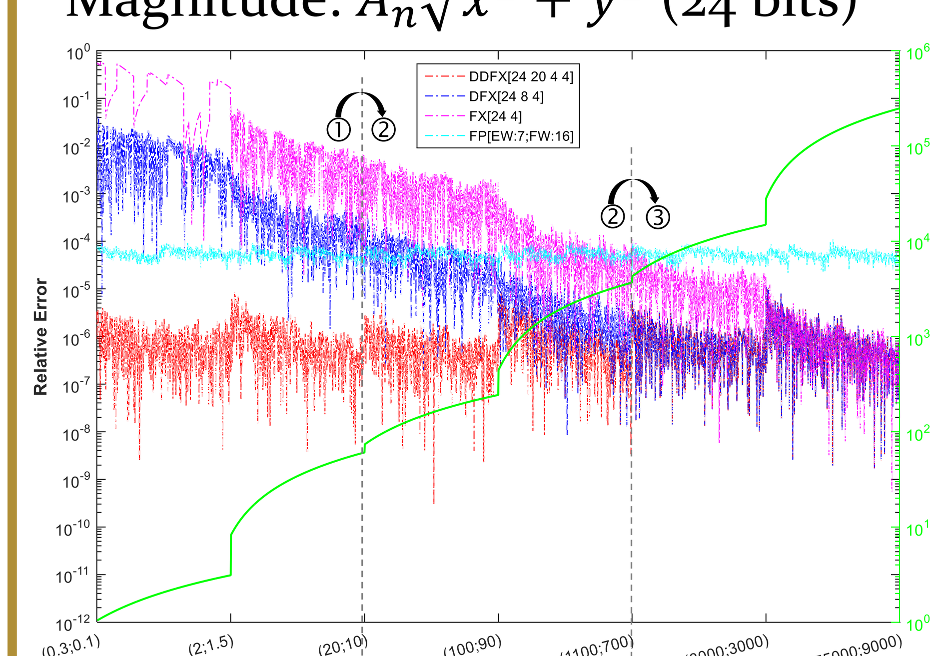
Floorplanning for the run-time alterable area increases resources (compared to DFX) up to 50% in the 32-bit case.

32-bit CORDIC	Slices	24-bit CORDIC	Slices
DDFX [32 26 4 6]	380	DDFX [24 20 4 4]	249
DFX [32 10 4]	293	DFX [24 8 4]	244
DDFX [32 29 1 8]	383	DDFX [24 23 1 6]	257
DFX [32 9 1]	292	DFX [24 7 1]	243
FX [32 4], [32 1]	151	FX [24 4], [24 1]	119
FP single precision	456	FP: EW:7, FW:16	340

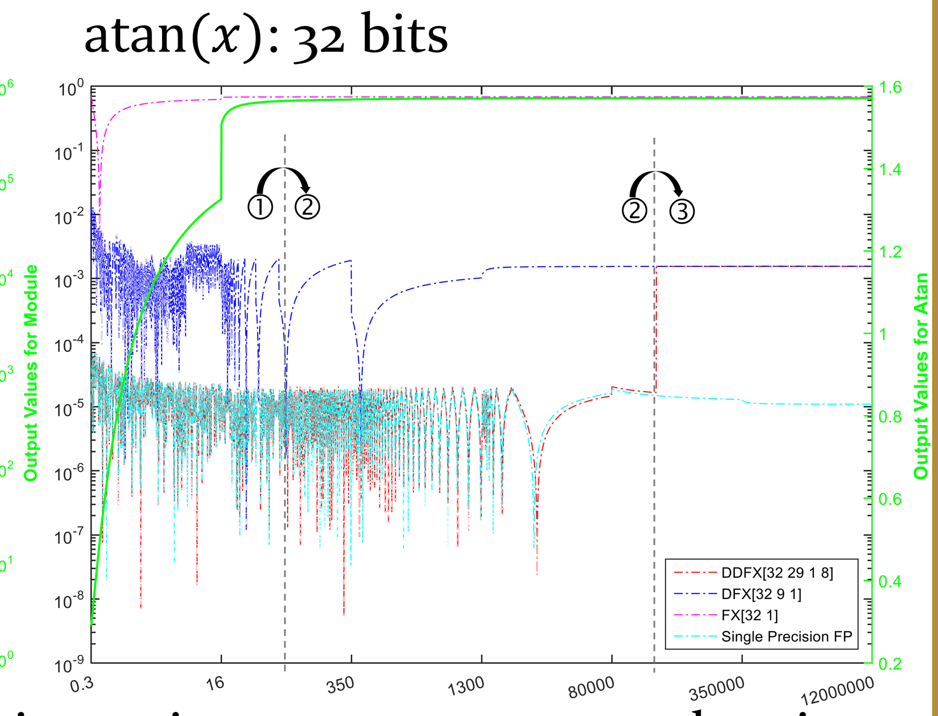
Accuracy (*Relative Error compared to MATLAB built-in functions*)

Increasing monotonic domains: See how states transition from 1 to 3.

Magnitude: $A_n\sqrt{x^2+y^2}$ (24 bits)

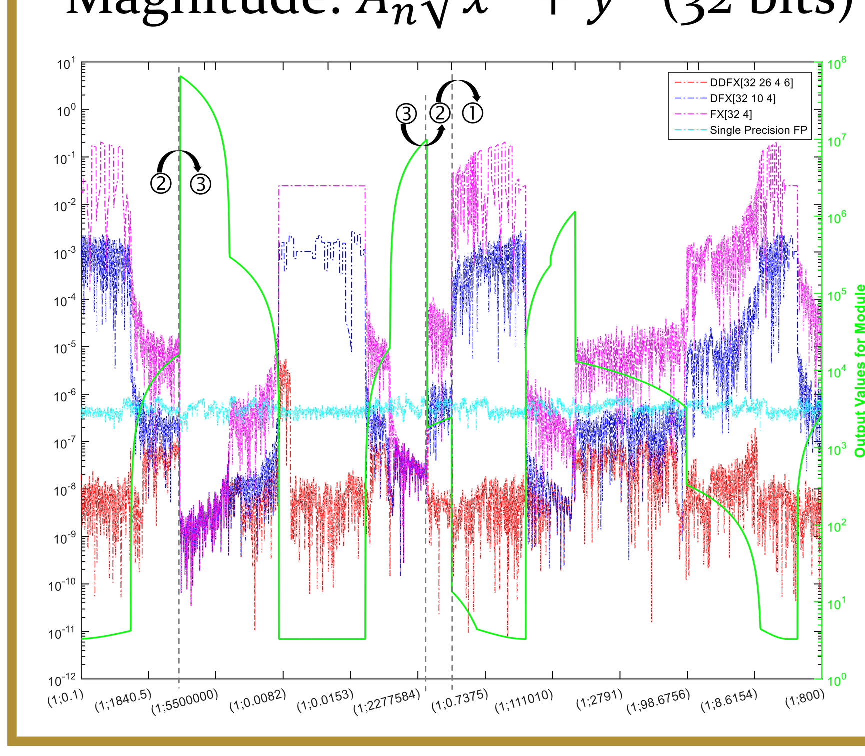


atan(x): 32 bits

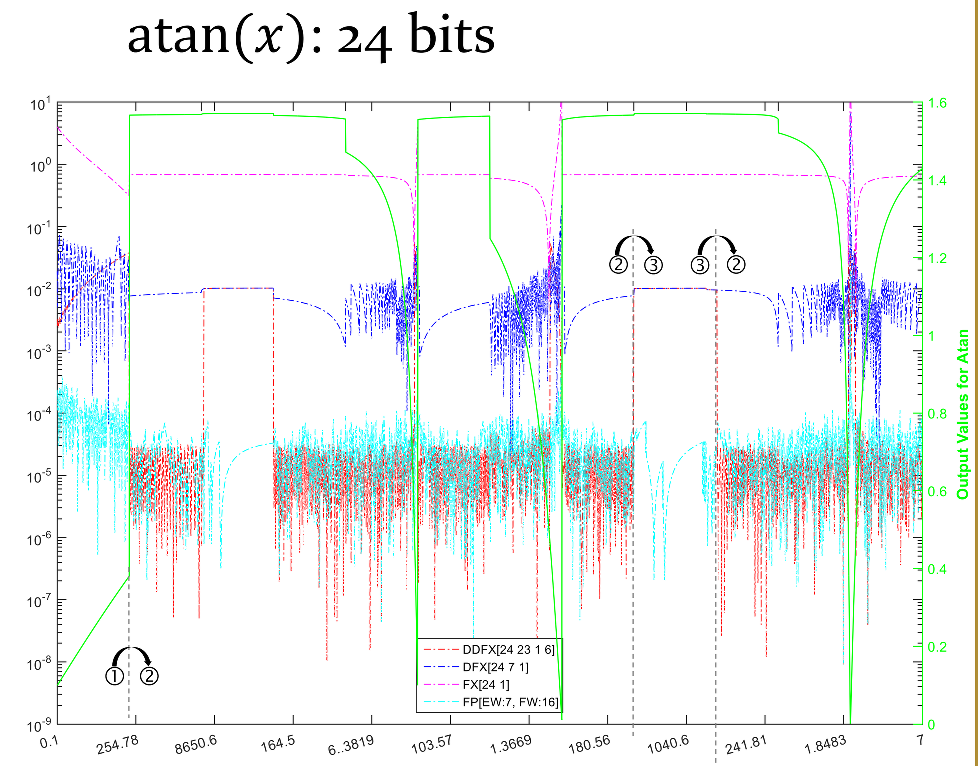


Non-monotonic domains: disjoint piece-wise vectors, even overlapping. This forces the State Diagram to transition back and forth between States 1 to 3. Note how DDFX is more accurate most of the time.

Magnitude: $A_n\sqrt{x^2+y^2}$ (32 bits)



atan(x): 24 bits



Conclusion

We presented and validated the circular DDFX CORDIC for magnitude and arc tangent. Comparisons with DFX, FX, and FP units suggest that DDFX is an efficient approach for CORDIC implementation: it allows for larger dynamic ranges than DFX, while delivering improved or very close accuracy over FP. The DDFX accuracy gains more than offset the hardware and software overhead with respect to DFX. Further work should optimize: i) the DDFX format to minimize overflow incidence and maximize accuracy, and ii) the run-time alterable logic (its routing caused low running frequencies and large reconfigurable sizes).