

## Low Power Trigonometric Calculation Using Simple Digital Logic

QIANG GAO

University of Strathclyde

[qiang.gao@eee.strath.ac.uk](mailto:qiang.gao@eee.strath.ac.uk)



# Summary

- Motivation
- Introduction
- CORDIC Fundamentals
- Parallel, “Pipelined Parallel”, and Serial CORDIC
- Barrel shifter based serial CORDIC design
- Methodology of low power design
- Conclusion

# Motivation

- In Specknet project, low power and complexity arithmetic unit plays an important role in 2D/3D motion
  - Rotation
  - Robotics

# Introduction

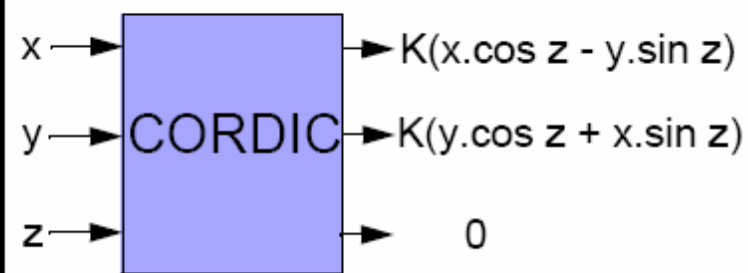
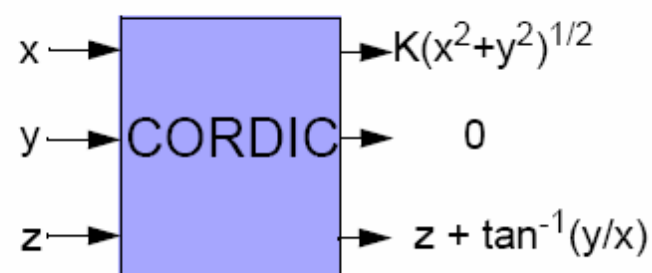
- Various trigonometric functions, such as sine, cosine and inverse tangent are widely required in 2D/3D motions.
- Implementing the low power arithmetic unit on digital logic becomes an important research area
- In this presentation the low complexity Coordinate Rotation Digital Computer (CORDIC) algorithm and its hardware implementation is discussed.

# Coordinate Rotation Digital Computer

- The Coordinate Rotation Digital Computer (CORDIC) algorithm is a simple technique using mainly shifts, additions and lookup tables (LUTs) – Ideal for hardware implementation
- Used to compute many different operations including trigonometric functions
- Based on Givens rotation
- Iterative technique

# CORDIC output

- The output from the CORDIC algorithm operating in Rotation & Vectoring modes is:

Coordinate System	Rotation Mode $z^{(i)} \rightarrow 0; d_i = \text{sign}(z^{(i)})$	Vectoring Mode $y^{(i)} \rightarrow 0; d_i = -\text{sign}(x^{(i)}y^{(i)})$
Circular	 <p>For <math>\cos z</math> &amp; <math>\sin z</math>, set <math>x = 1/K, y = 0</math></p>	 <p>For <math>\tan^{-1} y</math>, set <math>x = 1, z = 0</math></p>

- The 'K' value is the scaling factor, which can be removed by multiplying the result by  $1/K$ .

# Shift-Add algorithm

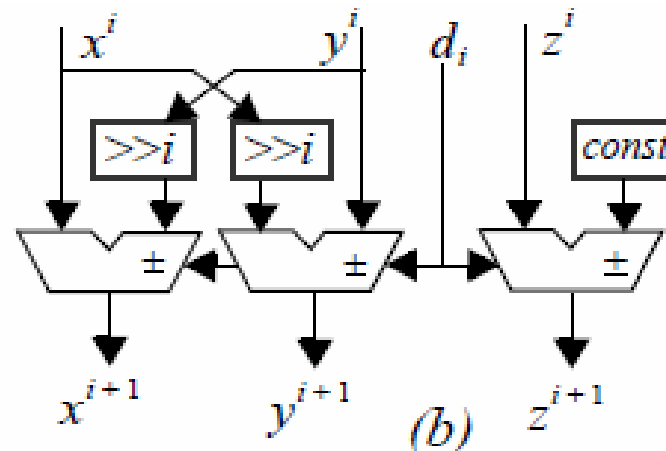
- Original Givens rotation now reduced to iterative shift-add algorithm
- Operations required are ideal for hardware implementation

$$x^{i+1} = x^i - d_i(2^{-i}y^i)$$

$$y^{i+1} = y^i + d_i(2^{-i}x^i)$$

$$z^{i+1} = z^i - d_i(\text{atan}2^{-i})$$

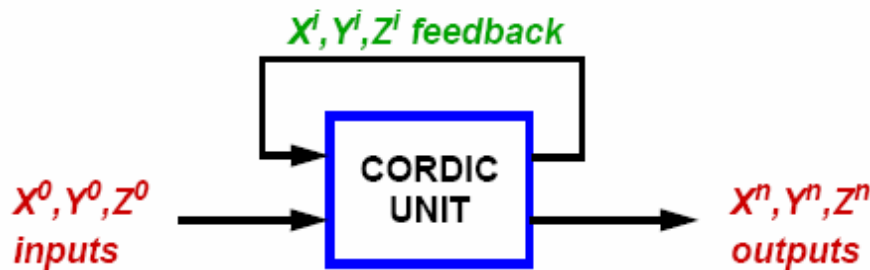
(a)



- Thus each iteration requires:
  - ❑ 2 shifts
  - ❑ 1 lookup table (values)
  - ❑ 3 additions

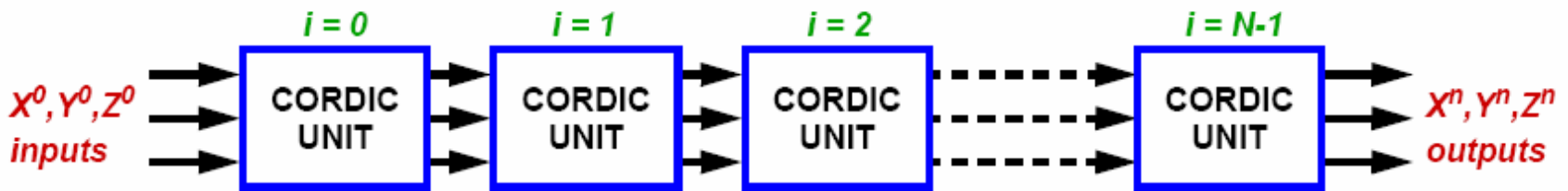
# Implementation Architectures

## 1) Rolled

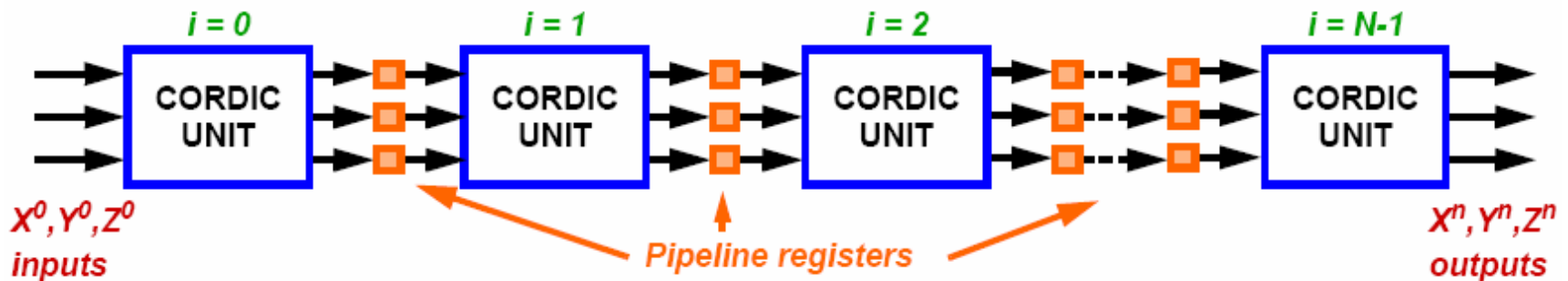


Note that the CORDIC unit in the rolled architecture is different compared to the unrolled designs: it must keep track of the iteration number at each clock cycle and synthesise the correct shift. As the CORDIC unit is shared  $N$  times, it runs at  $1/N$  of the rate.

## 2) Unrolled

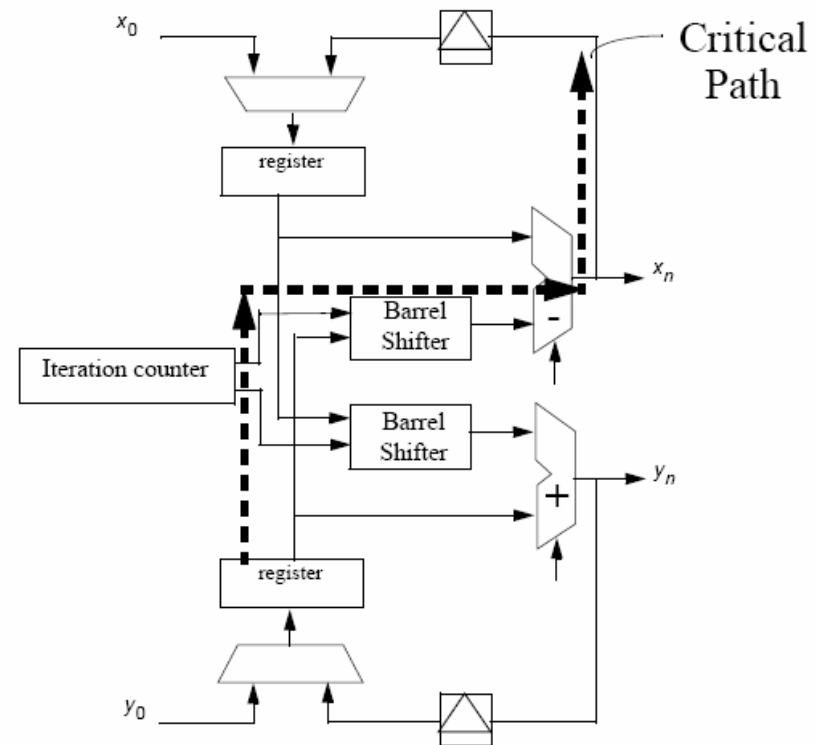


## 3) Unrolled and Pipelined



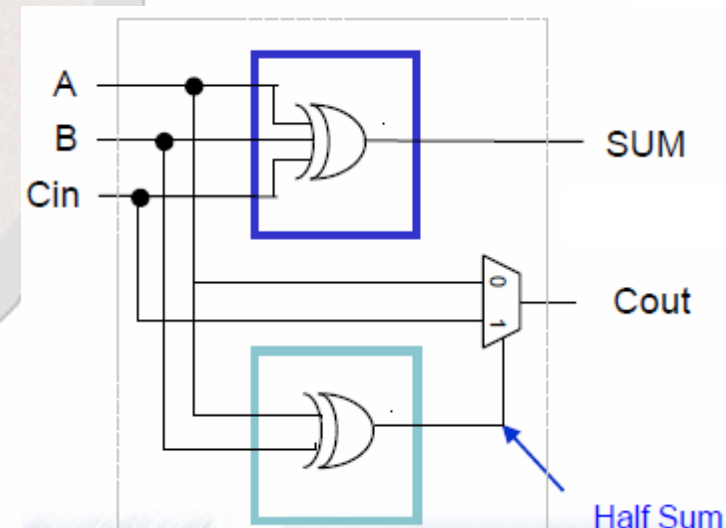
# Single Cell Based CORDIC

- Often a dedicated rolled architecture is used for each iteration. It can lower the hardware cost of the CORDIC processor by time-sharing.
- The main cost is the variable length shift register required.
- The barrel shifter is core part of the critical path.
- Pipelined Barrel Shifter will cause much higher speed



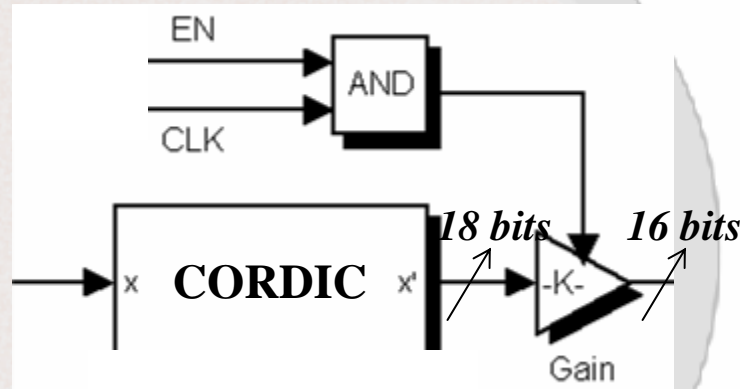
# Adder Implementation

- In VLSI technology the implementation of adders very much dependent on the performance required.
- There are three primary techniques in regard to adders : Ripple Carry Adder (RCA), Carry Lookahead Adder (CLA) and Carry Save Adder (CSA).
- RCA with Carry logic (right) is chosen in this work to utilize the lowest hardware, meanwhile maintain the high speed by avoiding the LUTs between Carry In and Carry Out.



# Low Power Implementation

- Apply Clock Gating technique on the Gain multiplier (below), which removes the scaling factor  $K$ .



- An Overall Quantization Error (OQE) estimation algorithm has been adopted to reduce the wordlength of Gain block (18 bits down to 16bits above).

# Hardware Implementation Result

- Xilinx Virtex 4 FPGA is used as the prototyping platform to implement three types of CORDIC architecture (in vector mode).
- The FPGA implementation result of serial CORDIC is listed as follows (under the iteration number = 18)

	<b>Cost</b>	<b>Sample Rate</b>	<b>Dynamic Power</b>
Serial CORDIC	275 Slices	10 MHz	0.00216 W

- So serial CORDIC architecture has very low power dissipation and hardware utilization.
- Fulfil the requirement of Specknet (1 time/sec - 100 times/sec)

# Conclusion

- Low complexity arithmetic unit plays as the wireless sensor network co-processor in Specknet system
- CORDIC algorithm has been presented
- The trade-off of Parallel, Pipelined Parallel and Serial CORDICs is discussed.
- The serial CORDIC implementation using pure digital logic has been described
- The pipelined Barrel Shifter based serial CORDIC suits to the low cost and power communication application in Specknet

# Questions

