

CORDIC Idea Multi-word Precision

20160125

Copyright (c) 2015 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Data Parallelism

Multi-word precision Multi-task

Objectives: High Precision, min area & power

Resource Sharing

Carry Propagate Network

Space-Time Optimization

Multi CORDIC with initial phase shifting

* parallelism to reduce the latency

* pipeline to increase the throughput

reduce the number of iterations
→ look ahead

loop unrolling

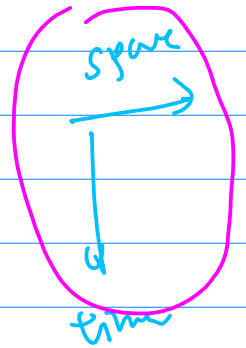
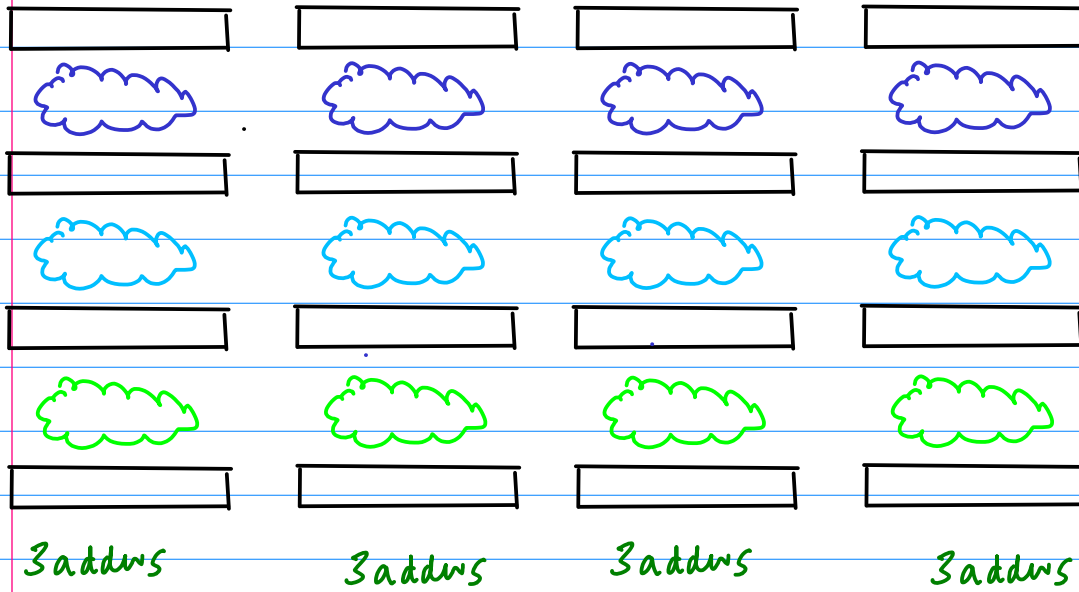
domino logic

without FF's nor latches

2 phase overlapping clocks



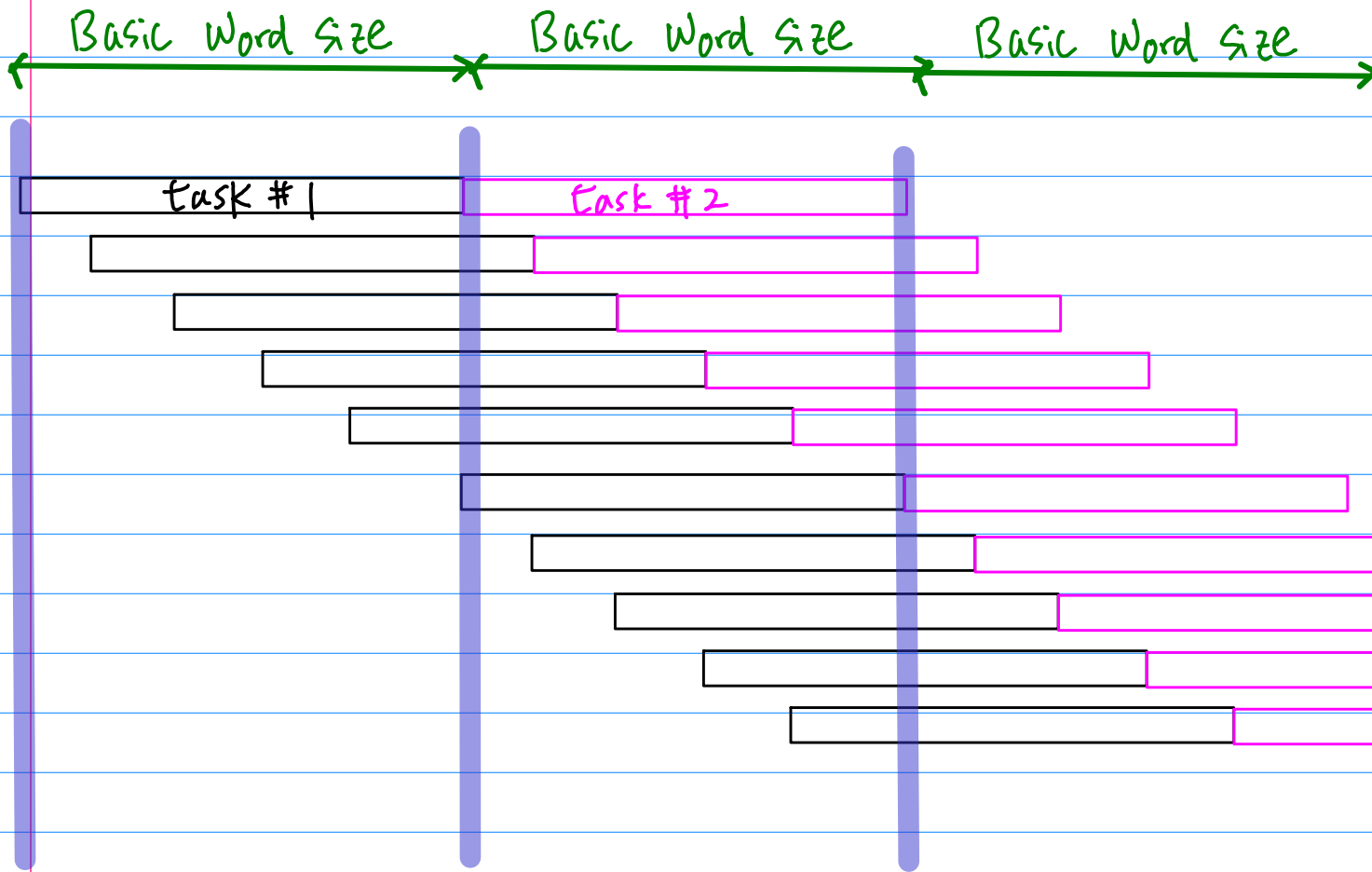
Data Parallelism



* serialization
of recurrence equation

maybe area-critical

Multi-word precision Multi-task

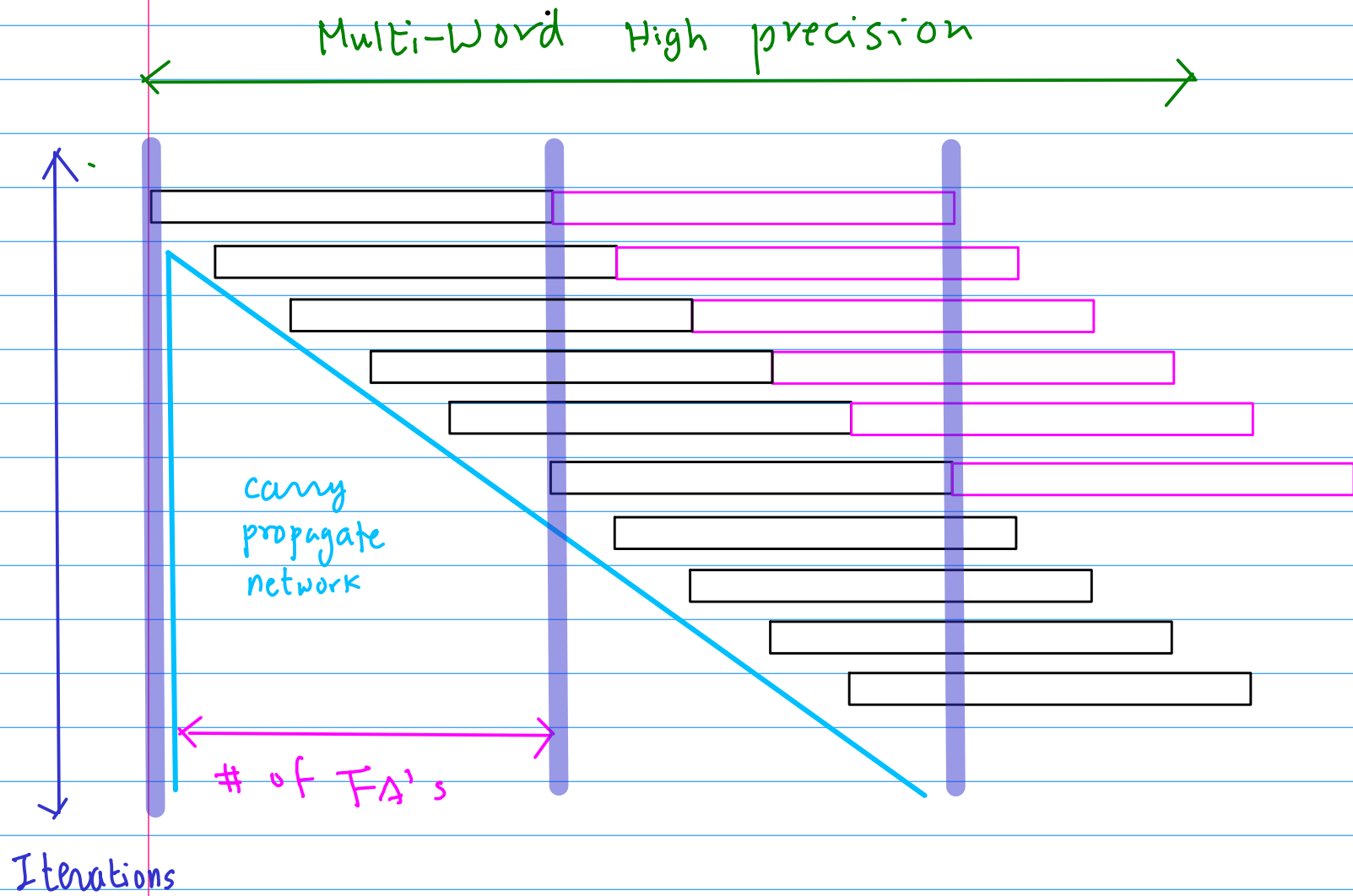


Multi-word precision
Multi-task

depending on the word size and iteration number

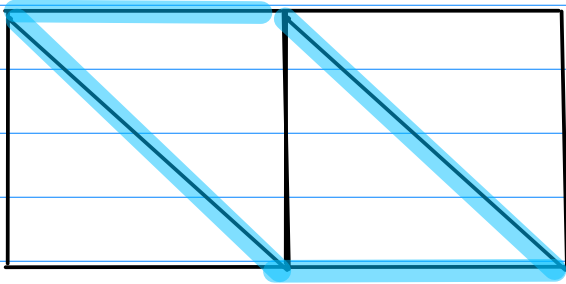
in each iteration, the same binary angle is
added / subtracted across different tasks / words

Objectives: High Precision, min area & power

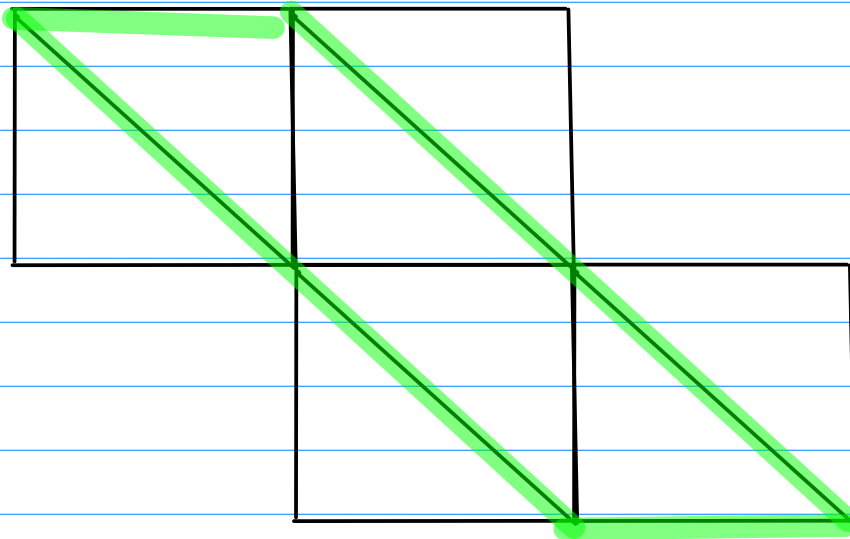


Optimize the Adder Circuits
Resource Sharing
Efficient Adder/Subtract

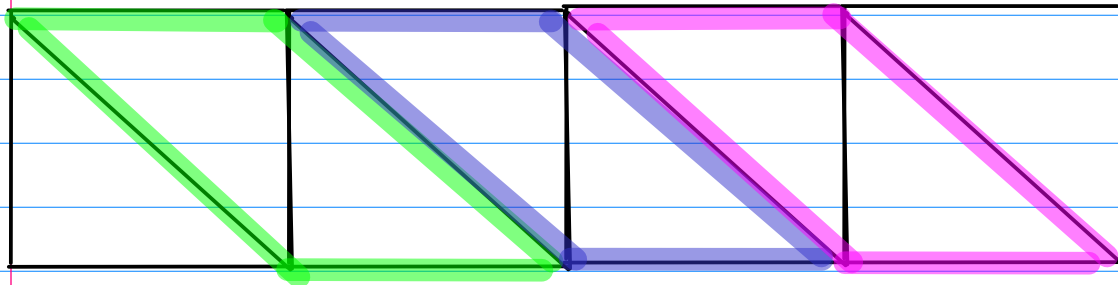
reduce area
reduce power



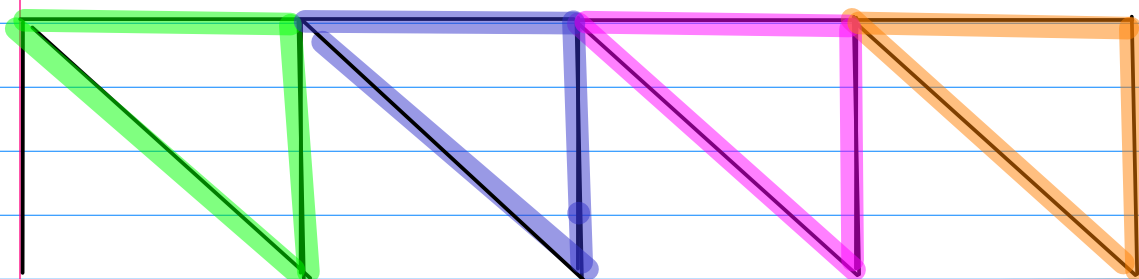
multi-word precision



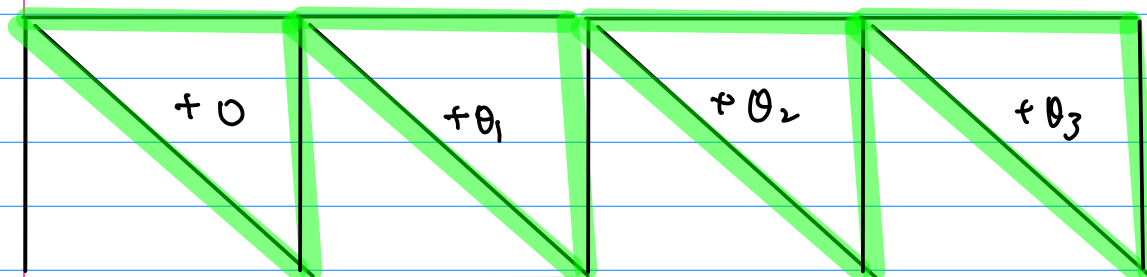
more high precision



multi-word precision
multi-tasks

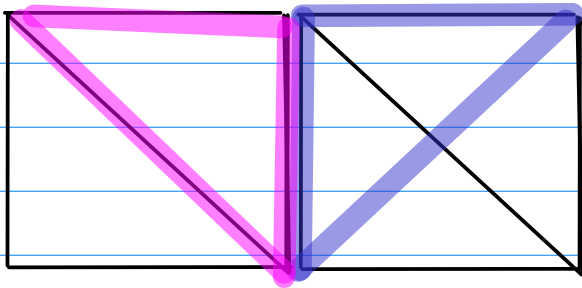
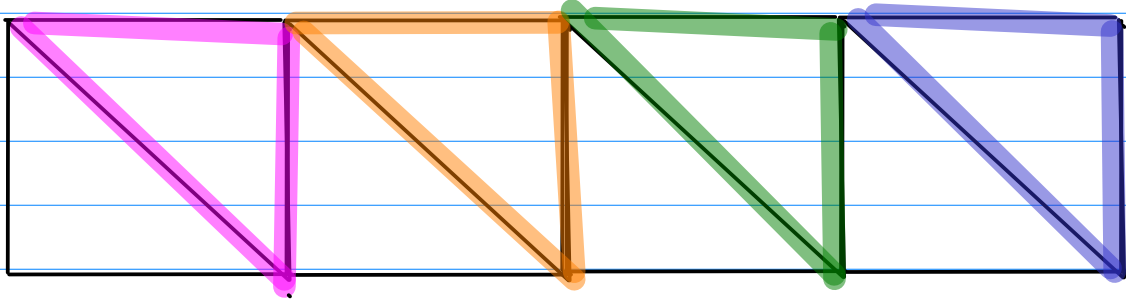
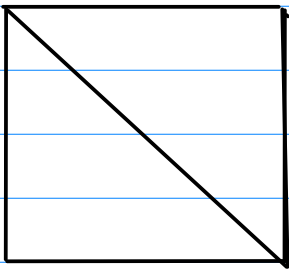


single-word precision
simple multi-tasks



single-task
Multi-CORDIC with initial phase shifting

Resource Sharing



at least
incrementer
is required

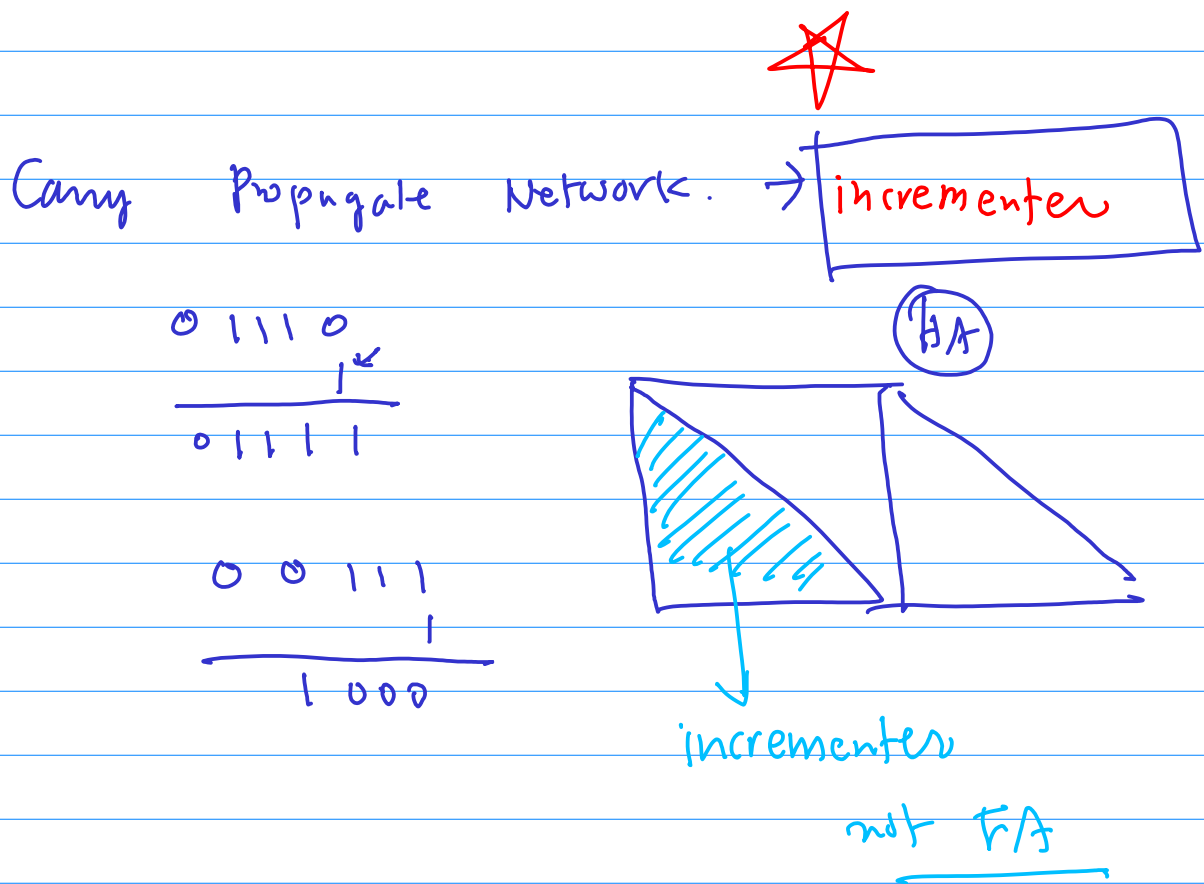
shifter
is required

↓
(HA)

less power

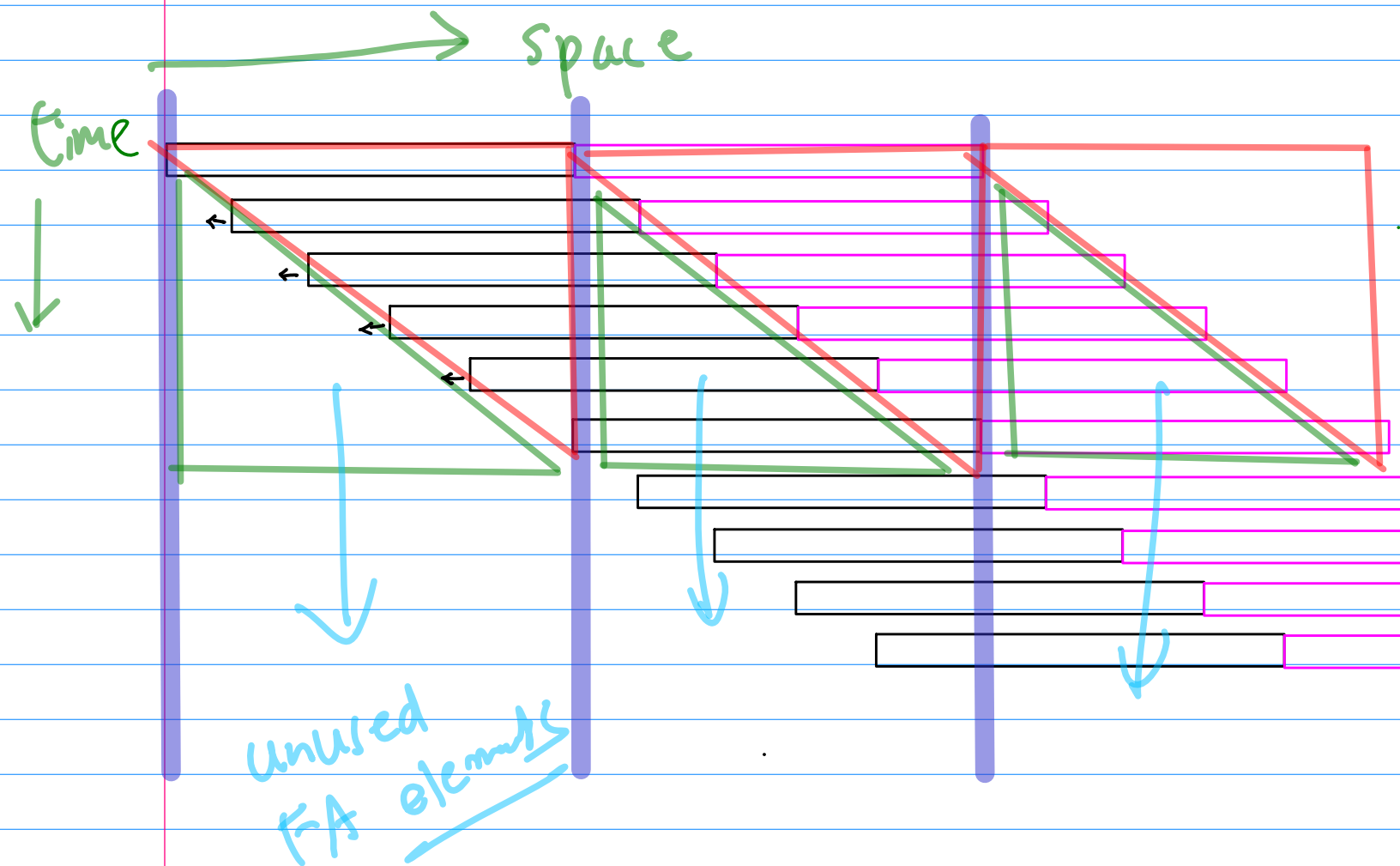
cant be used
in multi-precision

Carry Propagate Network



make sense only if it is
for a multi-word precision
or multi-task applications

Space - Time Optimization



Only if carry out propagate properly

FA's are used for propagating carries
no operand value.

space-time optimization

FA arrays
Registers for operands

x_i, y_i : all in the range $[-1, +1]$

separate hardware

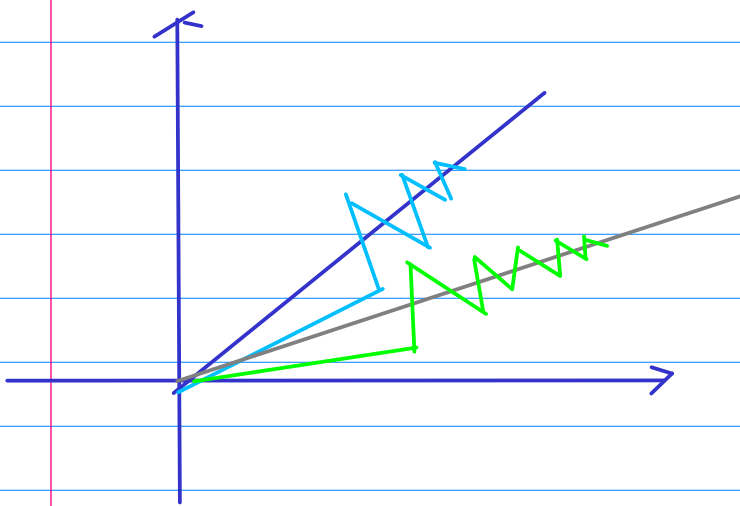


Multi CORDIC with initial phase shifting

if the searching angle falls into
dense area

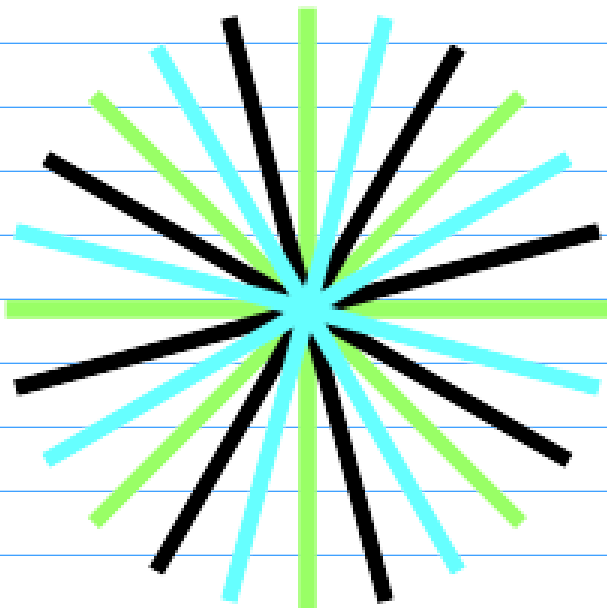
$0^\circ, 45^\circ, 90^\circ, 135^\circ,$

Then



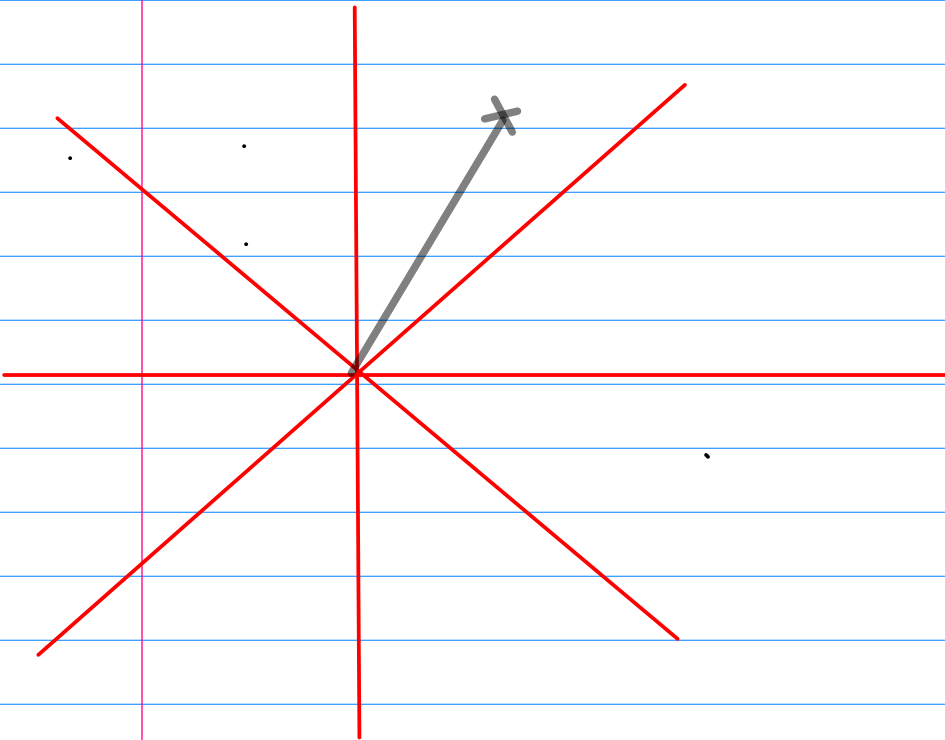
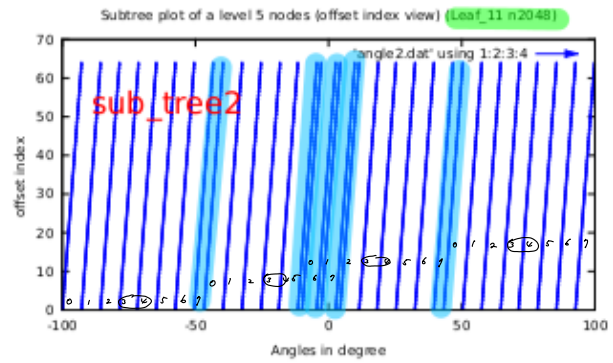
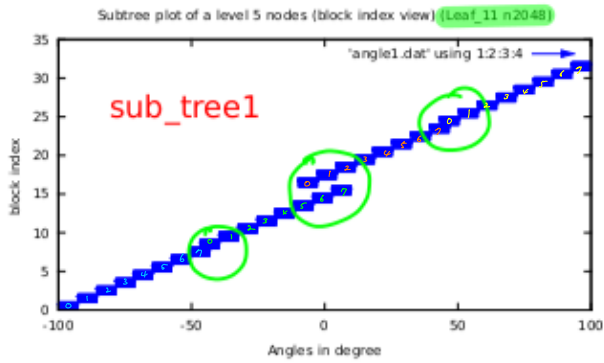
↖ pre shifting

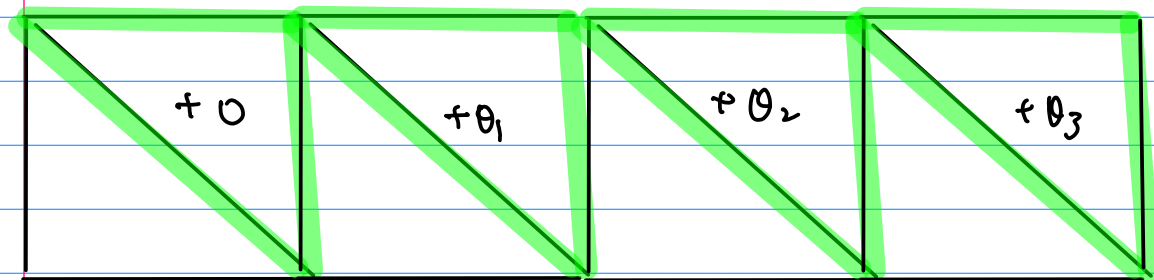
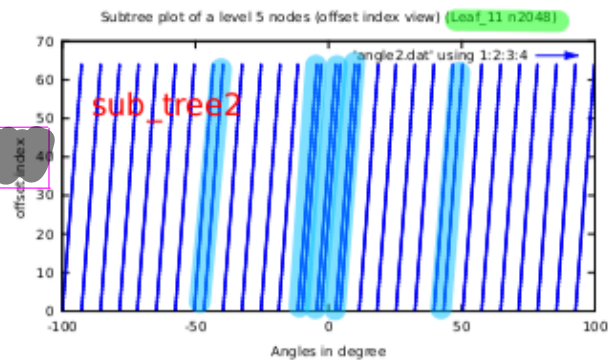
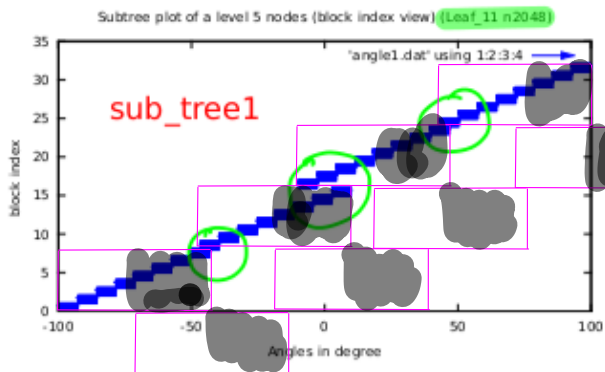
* multiple cordic engines



cordic iteration

First 3~4 steps : corresponds to "different slopes"





single-task
Multi-CORDIC with initial phase shifting

☆ So, is there a way to choose the best one to increase precision?

Heuristics ?

