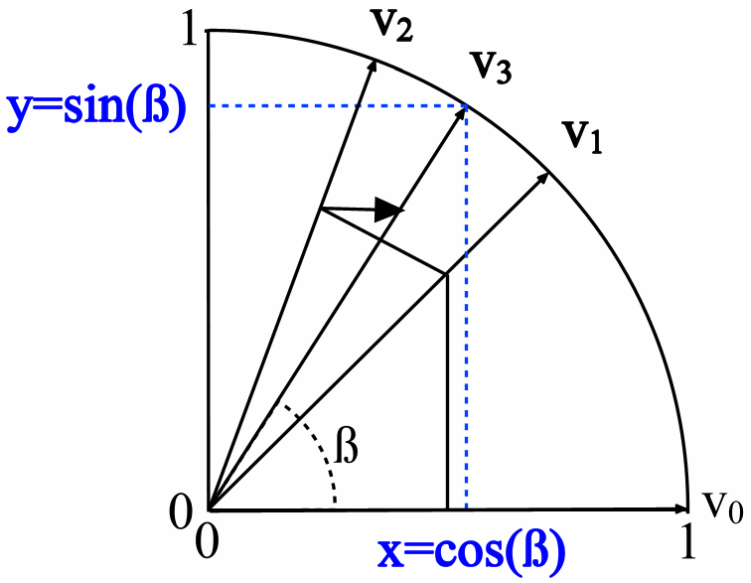
CORDIC-Based Trigonometric Functions

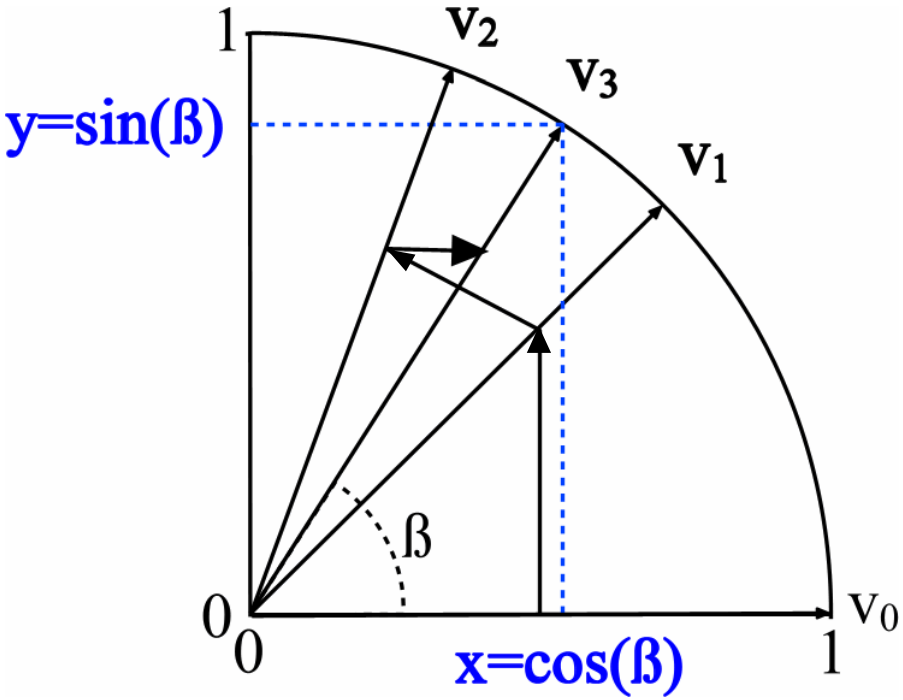
CH3 Computer Arithmetic
Programming Assignment

Prof. Ren-Shuo Liu
NTHU EE
Fall 2023



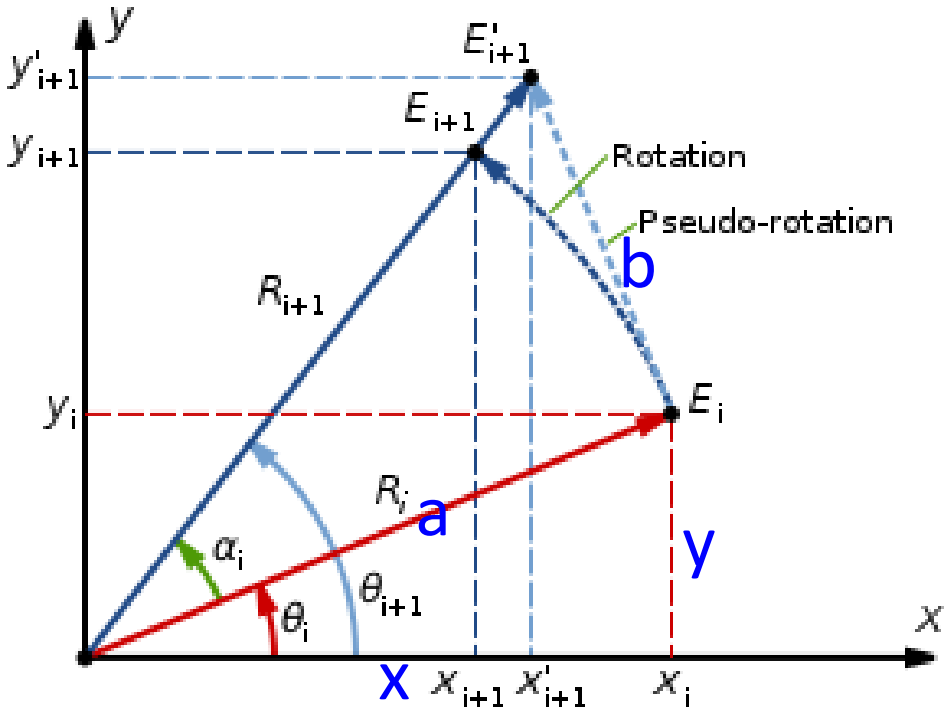
Basic Concept

- CORDIC is a binary approximation algorithm to calculate sin(β) and cos(β) using only integer arithmetic



Background

- (**x** + **y** i) × (**a** + **b** i) = (ax - by) + (bx + ay) i
  - Rotate the point by $\tan^{-1}(\frac{b}{a})$
  - Increase the distance by $\sqrt{\left(1+\left(\frac{b}{a}\right)^2\right)}$



CORDIC

- a = 1
- b = 2^{-k}
- Only shift and addition
  - (**x** + **y** i) × (**a** + **b** i)
    = (ax - by) + (bx + ay) i
    = (**x** - (y>>**k**)) + ((**x**>>k) + **y**)
- $\tan^{-1}$(b/a) is precomputed and stored in a table

a	**b**	**$\tan^{-1}$(b/a)**
1	1	45°
1	0.5	26.5651…°
1	2^{-2}	14.0362…°
1	…	…
1	$2^{-(N-1)}$	…

Software Implementation

```
x = initial_x;
y = 0;
N = # iterations, 20 (fixed);
th = the angle to compute;
for k = 0 … (N – 1)
   if th >= 0
     (x, y) *= (1, 2^-k);
     th = th - tan^-1(2^-k);
   else
     (x, y) *= (1, -2^-k);
     th = th + tan^-1(2^-k);

return x as cos(th)
   and y as sin(th);
```

a	b	$\tan^{-1}(b/a)$
1	1	45°
1	0.5	26.5651…°
1	2^{-2}	14.0362…°
1	…	…
1	$2^{-(N-1)}$	…

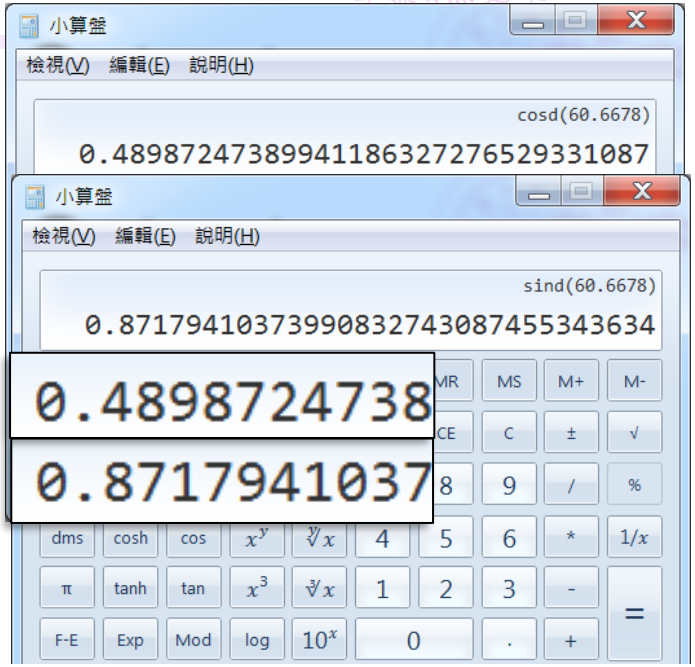
Example Input and Output

- Input

```
5
606678
457006
-837565
-835975
395400
```

// number of inputs
// 60.6678° (scaled by 10,000)
// 45.7006° (scaled by 10,000)

note: for simplicity,
input < 90°
&& input > -90°
&& input != 0°



- Output

```
5
606678 4898750 8717926
457006 6984117 7156964
-837565 1087524 -9940686
-835975 1115127 -9937628
395400 7711808 6366160
```

60.6678°
cos(60.6678°) = 0.4898751 (scaled by 10,000,000)
sin(60.6678°) = 0.8717926 (scaled by 10,000,000)

Note: use **scanf/printf** or **cin/cout** to handle the input/output

Parameters for this Assignment

k	$\tan^{-1}(2^{-k})$
0	450000
1	265651
2	140362
3	71250
4	35763
5	17899
6	8952
7	4476
8	2238
9	1119

k	$\tan^{-1}(2^{-k})$
10	560
11	280
12	140
13	70
14	35
15	17
16	9
17	4
18	2
19	1

initial_x = **6072529**;

Template

- Template is provided at: https://github.com/JerryWang0520/ee3450_pa2

- Please refer to **README.md** to check he correctness.

- Example input and output with 100 testcases are provided as reference.

Delivery

- Rename your **main.c** (**main.cpp**) as
  - **PA2_<student_ID>.c** (**PA2_<student_ID>.cpp**)

  For example: **PA2_109061585.c** (**PA2_109061585.cpp**)

- You can choose either **C** or **C++** to finish.

- Send your code through eeclass.

Hardware (Just for Reference)