

## Código Assembly -> CORDIC: Luis Felipe de Deus, Pedro Basso, Lucas Lauck

#Register Map

#t0 - Angle

#t1 - It

#t2 - x

#t3 - y

#t4 - xnew

#t5 - ynew

#t6 - i

#t7 - sumAngle

#t8 - j

#t9 - temp

#s5 - angleTable[i]

.text

cordic:

addi \$t0, \$t0, 90 #Carrega dado no reg angle

sll \$t0, \$t0, 24 #Desloca reg angle

addi \$t2, \$zero, 0x9B74ED #Carrega valor inicial de x

addi \$t1, \$t1, 32 #Carrega valor de iterações

for:

slt \$t9, \$t6, \$t1 #Se i < it t9 recebe 1

beq \$t9, \$zero, fim #Se t9 for igual a zero pula pro fim

Shift:

sra \$s6, \$t2, \$t6 #Shift aritmético do x ( $x \gg i$ )

sra \$s7, \$t3, \$t6 #Shift aritmético do y ( $y \gg i$ )

mem:

la \$s5, angleTable #Carrega o endereço inicial do array

add \$s5, \$s5, \$t8 #Soma com o endereço atual de i

lw \$s5, (\$s5) #Carrega o valor do angleTable[i]

teste:

slt \$t9,\$t7,\$t0 #Se sumAngle < angle t9 recebe 1 (vai ir pro if )

beq \$t9, \$zero, else # se t9 = zero vai pro else, angle menor que ou igual a sumAngle

if:

sub \$t4, \$t2, \$s7 #x\_new = x - (y>>i);

add \$t5, \$t3, \$s6 #y\_new = y + (x>>i);

add \$t7, \$t7, \$s5 #sumAngle = sumAngle + angleTable[i]

j fimfor #Salto para o fim do for, não executa else

else:

add \$t4, \$t2, \$s7 #x\_new = x + (y>>i);

sub \$t5, \$t3, \$s6 #y\_new = y - (x>>i);

sub \$t7, \$t7, \$s5 #sumAngle = sumAngle - angleTable[i]

fimfor:

add \$t2, \$t4, \$zero #x <- xnew

add \$t3, \$t5, \$zero #y <- ynew

addi \$t6, \$t6, 1 #i++;

addi \$t8, \$t8, 4 #j+=4 Pra endereçar a memoria (words)

j for #Loop

fim:

add \$s0, \$zero, \$t2 #Salva valor em S0 -> Cosseno

add \$s1, \$zero, \$t3 #Salva valor em S1 -> Seno

#dados

.data

size: .word 32

angleTable: .word 0x2d000000 0x1a90a731 0x0e094740 0x07200112 0x03938aa6 0x01ca3794

0x00e52a1a 0x007296d7 0x00394ba5 0x001ca5d9 0x000e52ed 0x00072976

0x000394bb 0x0001ca5d 0x0000e52e 0x00007297 0x0000394b 0x00001ca5

0x00000e52 0x00000729 0x00000394 0x000001ca 0x000000e5 0x00000072

0x00000039 0x0000001c 0x0000000e 0x00000007 0x00000003 0x00000001

0x00000000 0x00000000

TEMPOS DOS PROCESSADORES:

EM CICLOS DE CLOCK COM PERIODO DE 4 ns -> Frequência 250 MHz

\*Testes realizados com ângulo de 75º com 32 iterações para todos processadores

Processador 3: MIPS + CORDIC -> 2250 CICLOS DE CLOCK

Processador 2: CORDIC ESTRUTUAL + COMPORTAMENTAL -> 197 CICLOS DE CLOCK

Processador 1: CORDIC TOTALMENTE COMPORTAMENTAL -> 69 CICLOS DE CLOCK