



Universidade de Brasília
Departamento de Ciência da Computação

Universidade de Brasília (UnB)

CIC0099 – Organização e Arquitetura de Computadores –
Unificado 2025/2

Relatório Laboratório 1 – OAC

Grupo A4:

Artur Nunes de Souza - Matrícula: 241002485

Eduardo Oliveira Vieira - Matrícula: 242010024

Filipe Nobre Nunes - Matrícula: 242023523

Guilherme Silva Cavalcante - Matrícula: 242001508

Rafael Mileo Moreira Krauss Guimarães - Matrícula: 242024763

Sumário

0.1	Item 1	2
	0.1.1 Item 1.2	2
0.2	Item 2	5
0.3	Item 3	16

0.1 Item 1

0.1.1 Item 1.2

Feito considerando a inversão da ordenação realizada no tópico 1.1.

a)

A equação de tempo de execução é dada por $I \times CPI \times T$, neste caso, é considerado que $CPI = 1$ e $T = \frac{1}{50.000.000}$, o inverso da frequência dada. Logo, $I \times \frac{1}{50.000.000}$ é o tempo de execução base, dependendo das instruções que serão executadas, que depende de n , o tamanho do array, e do modo em que ele está ordenado.

Considerando que neste exemplo um está ordenado crescentemente e o outro decrescentemente, basta calcular o número de instruções que serão executadas em cada iteração, levando em conta n em cada um dos dois casos e substituir por I na fórmula acima. Além disso, é necessário somar 1 às instruções, dado que o breakpoint ocorre antes do retorno. Por fim, a função da quantidade de instruções dado n será denotada por $S(n)$.

A partir de testes ($S(3) = 86, S(4) = 145$), foi possível encontrar a fórmula a partir dos seguintes cálculos:

Cálculo a partir das diferenças segundas

Primeiro, foram calculadas as diferenças entre valores consecutivos:

Diferenças de primeira ordem (ΔS):

$$\Delta S = [45 - 22, 86 - 45, 145 - 86, 222 - 145] = [23, 41, 59, 77]$$

Depois, as diferenças entre essas diferenças (segunda ordem):

$$\Delta^2 S = [41 - 23, 59 - 41, 77 - 59] = [18, 18, 18]$$

As segundas diferenças são **constantes ou seja $S(n)$ é quadrática**:

$$S(n) = a \cdot n^2 + b \cdot n + c$$

Usando três pontos conhecidos ($n = 1, 2, 3$):

$$\begin{cases} a + b + c = 22 \\ 4a + 2b + c = 45 \\ 9a + 3b + c = 86 \end{cases}$$

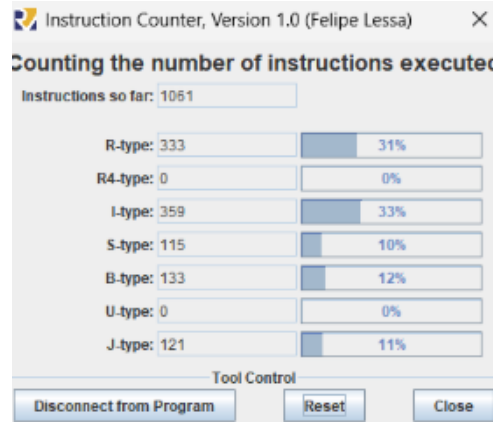
A partir desse sistema de equações, é possível chegar na fórmula:

$$S(n) = 9n^2 - 4n + 17$$

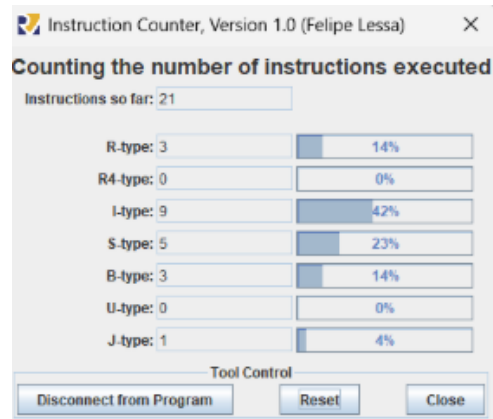
Com $to(n)$ (Caso com o array ordenado crescentemente):

$$t_{exec}(to) = \frac{9n^2 - 4n + 17}{50.000.000}$$

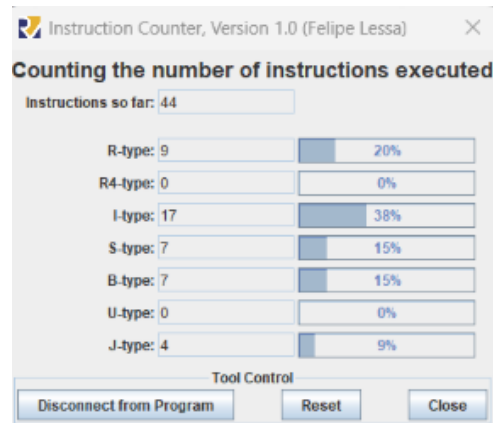
com $S(11)$, $t_{exec} = 2.124 \times 10^{-5}$



com $S(1)$, $t_{exec} = 4.4 \times 10^{-7}$

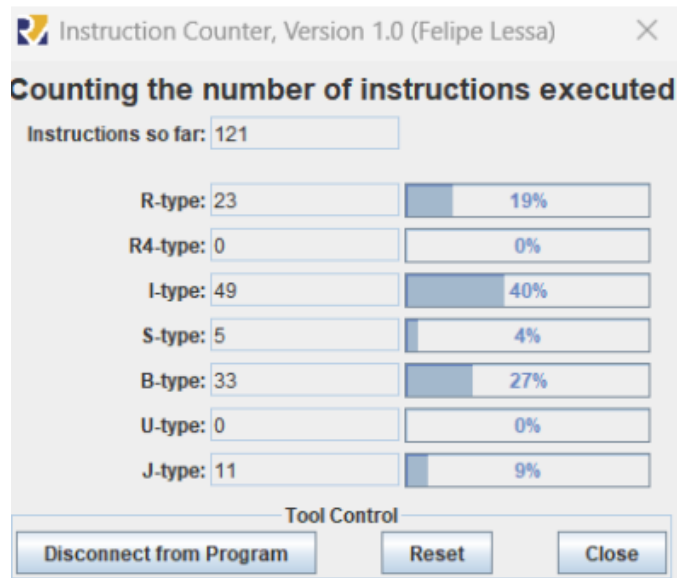


com $S(2)$, $t_{exec} = 9 \times 10^{-5}$

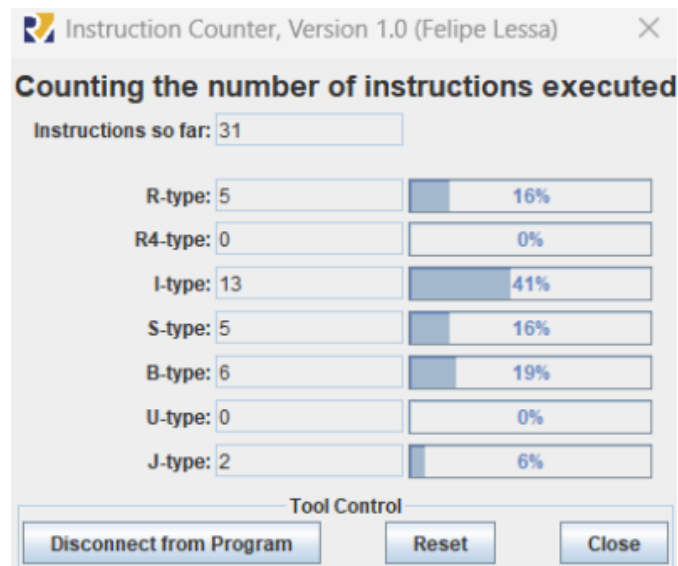


Similarmente, um cálculo das diferenças foi realizado para encontrar o número de instruções do procedimento sort. Foi notado que a quantidade de instruções em função de n era ditado pela seguinte equação: $12 + 10n$, de modo que o $t_{exec}(to) = \frac{12+10.n}{50.000.000}$

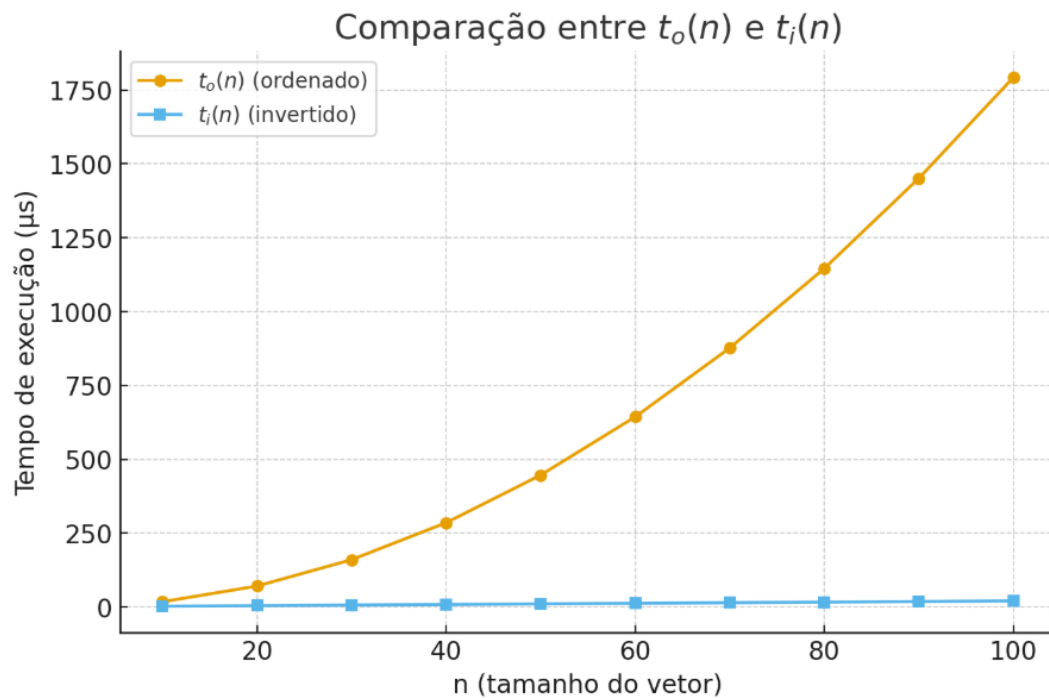
Para S(11):



Para S(2):



b) Por ser uma função quadrática o crescimento de $t_o(n)$ é muito mais acentuado que



o de $t_i(n)$, o que pode ser visto no gráfico.

0.2 Item 2

2.2 O código, ao ser compilado conforme o item gera o seguinte:

```

1  v :
2      .word    9
3      .word    2
4      .word    5
5      .word    1
6      .word    8
7      .word    2
8      .word    4
9      .word    3
10     .word    6
11     .word    7
12     .word    10
13     .word    2
14     .word    32
15     .word    54
16     .word    2
17     .word    12
18     .word    6

```

```

19         .word    3
20         .word    1
21         .word    78
22         .word    54
23         .word    23
24         .word    1
25         .word    54
26         .word    2
27         .word    65
28         .word    3
29         .word    6
30         .word    55
31         .word    31
32 .LC0:
33         .string  "%d\t"
34 show(int*, int):
35         addi     sp,sp,-48
36         sw      ra,44(sp)
37         sw      s0,40(sp)
38         addi     s0,sp,48
39         sw      a0,-36(s0)
40         sw      a1,-40(s0)
41         sw      zero,-20(s0)
42         j       .L2
43 .L3:
44         lw      a5,-20(s0)
45         slli     a5,a5,2
46         lw      a4,-36(s0)
47         add      a5,a4,a5
48         lw      a5,0(a5)
49         mv      a1,a5
50         lui      a5,%hi(.LC0)
51         addi     a0,a5,%lo(.LC0)
52         call     printf
53         lw      a5,-20(s0)
54         addi     a5,a5,1
55         sw      a5,-20(s0)
56 .L2:
57         lw      a4,-20(s0)
58         lw      a5,-40(s0)
59         blt      a4,a5,.L3

```

```

60      li      a0,10
61      call    putchar
62      nop
63      lw      ra,44(sp)
64      lw      s0,40(sp)
65      addi     sp,sp,48
66      jr      ra
67 swap(int*, int):
68      addi     sp,sp,-48
69      sw      ra,44(sp)
70      sw      s0,40(sp)
71      addi     s0,sp,48
72      sw      a0,-36(s0)
73      sw      a1,-40(s0)
74      lw      a5,-40(s0)
75      slli     a5,a5,2
76      lw      a4,-36(s0)
77      add      a5,a4,a5
78      lw      a5,0(a5)
79      sw      a5,-20(s0)
80      lw      a5,-40(s0)
81      addi     a5,a5,1
82      slli     a5,a5,2
83      lw      a4,-36(s0)
84      add      a4,a4,a5
85      lw      a5,-40(s0)
86      slli     a5,a5,2
87      lw      a3,-36(s0)
88      add      a5,a3,a5
89      lw      a4,0(a4)
90      sw      a4,0(a5)
91      lw      a5,-40(s0)
92      addi     a5,a5,1
93      slli     a5,a5,2
94      lw      a4,-36(s0)
95      add      a5,a4,a5
96      lw      a4,-20(s0)
97      sw      a4,0(a5)
98      nop
99      lw      ra,44(sp)
100     lw      s0,40(sp)

```



```

101         addi    sp,sp,48
102         jr      ra
103 sort(int*, int):
104         addi    sp,sp,-48
105         sw      ra,44(sp)
106         sw      s0,40(sp)
107         addi    s0,sp,48
108         sw      a0,-36(s0)
109         sw      a1,-40(s0)
110         sw      zero,-20(s0)
111         j       .L6
112 .L10:
113         lw      a5,-20(s0)
114         addi    a5,a5,-1
115         sw      a5,-24(s0)
116         j       .L7
117 .L9:
118         lw      a1,-24(s0)
119         lw      a0,-36(s0)
120         call    swap(int*, int)
121         lw      a5,-24(s0)
122         addi    a5,a5,-1
123         sw      a5,-24(s0)
124 .L7:
125         lw      a5,-24(s0)
126         blt     a5,zero,.L8
127         lw      a5,-24(s0)
128         slli    a5,a5,2
129         lw      a4,-36(s0)
130         add     a5,a4,a5
131         lw      a4,0(a5)
132         lw      a5,-24(s0)
133         addi    a5,a5,1
134         slli    a5,a5,2
135         lw      a3,-36(s0)
136         add     a5,a3,a5
137         lw      a5,0(a5)
138         bgt     a4,a5,.L9
139 .L8:
140         lw      a5,-20(s0)
141         addi    a5,a5,1

```

```

142         sw      a5,-20(s0)
143 .L6:
144         lw      a4,-20(s0)
145         lw      a5,-40(s0)
146         blt     a4,a5,.L10
147         nop
148         nop
149         lw      ra,44(sp)
150         lw      s0,40(sp)
151         addi    sp,sp,48
152         jr      ra
153 main:
154         addi    sp,sp,-16
155         sw      ra,12(sp)
156         sw      s0,8(sp)
157         addi    s0,sp,16
158         li      a1,30
159         lui     a5,%hi(v)
160         addi    a0,a5,%lo(v)
161         call    show(int*, int)
162         li      a1,30
163         lui     a5,%hi(v)
164         addi    a0,a5,%lo(v)
165         call    sort(int*, int)
166         li      a1,30
167         lui     a5,%hi(v)
168         addi    a0,a5,%lo(v)
169         call    show(int*, int)
170         li      a5,0
171         mv      a0,a5
172         lw      ra,12(sp)
173         lw      s0,8(sp)
174         addi    sp,sp,16
175         jr      ra

```

Do modo como está ele não pode ser executado diretamente no RARS, de modo que algumas mudanças são necessárias, principalmente relacionadas a como as funções estão sendo “chamadas”, na estrutura de algumas e na função main e em seu posicionamento. Código após as transformações:

```

1 .data

```

```

2  v:
3  .word
   9,2,5,1,8,2,4,3,6,7,10,2,32,54,2,12,6,3,1,78,54,23,1,54,2,65,3,6,55,31
4
5  .text
6
7  main:  #alterada para funcionamento e levada para cima para ser
        executada primeiro
8      addi    sp,sp,-16
9      sw      ra,12(sp)
10     sw      s0,8(sp)
11     addi    s0,sp,16
12     li      a1,30
13     la      a0,v
14     jal     show          #call mudado para jal
15     li      a1,30
16     la      a0,v
17     jal     sort          #call mudado para jal
18     li      a1,30
19     la      a0,v
20     jal     show          #call mudado para jal
21     lw      ra,12(sp)
22     lw      s0,8(sp)
23     addi    sp,sp,16
24     li      a7,10
25     ecall
26
27  show:          # substitui "show(int*, int)"
28     addi    sp,sp,-48
29     sw      ra,44(sp)
30     sw      s0,40(sp)
31     addi    s0,sp,48
32     sw      a0,-36(s0)
33     sw      a1,-40(s0)
34     sw      zero,-20(s0)
35     j       L2
36  L3:
37     lw      a5,-20(s0)
38     slli    a5,a5,2
39     lw      a4,-36(s0)

```

```

40      add      a5,a4,a5
41      lw       a5,0(a5)
42
43      # --- substitui "printf("%d\t", v[i])" ---
44
45      mv       a0,a5          # valor a ser imprimido
46      li       a7,1
47      ecall
48      li       a0,9          # TAB ap s o valor
49      li       a7,11
50      ecall
51
52      # -----
53
54      lw       a5,-20(s0)
55      addi     a5,a5,1
56      sw       a5,-20(s0)
57 L2:
58      lw       a4,-20(s0)
59      lw       a5,-40(s0)
60      blt      a4,a5,L3
61
62      # --- substitui "putchar" -----
63      li       a0,10
64      li       a7,11
65      ecall
66      # -----
67
68      lw       ra,44(sp)
69      lw       s0,40(sp)
70      addi     sp,sp,48
71      jr       ra
72
73
74 swap: #substitui "swap(int*, int)"
75      addi     sp,sp,-48
76      sw       ra,44(sp)
77      sw       s0,40(sp)
78      addi     s0,sp,48
79      sw       a0,-36(s0)
80      sw       a1,-40(s0)

```

```

81      lw      a5,-40(s0)
82      slli    a5,a5,2
83      lw      a4,-36(s0)
84      add     a5,a4,a5
85      lw      a5,0(a5)
86      sw      a5,-20(s0)
87      lw      a5,-40(s0)
88      addi    a5,a5,1
89      slli    a5,a5,2
90      lw      a4,-36(s0)
91      add     a4,a4,a5
92      lw      a5,-40(s0)
93      slli    a5,a5,2
94      lw      a3,-36(s0)
95      add     a5,a3,a5
96      lw      a4,0(a4)
97      sw      a4,0(a5)
98      lw      a5,-40(s0)
99      addi    a5,a5,1
100     slli    a5,a5,2
101     lw      a4,-36(s0)
102     add     a5,a4,a5
103     lw      a4,-20(s0)
104     sw      a4,0(a5)
105     #nop
106     lw      ra,44(sp)
107     lw      s0,40(sp)
108     addi    sp,sp,48
109     jr      ra
110
111 sort: #substitui "sort(int*, int)"
112     addi    sp,sp,-48
113     sw      ra,44(sp)
114     sw      s0,40(sp)
115     addi    s0,sp,48
116     sw      a0,-36(s0)
117     sw      a1,-40(s0)
118     sw      zero,-20(s0)
119     j       L6
120 L10:
121     lw      a5,-20(s0)

```

```

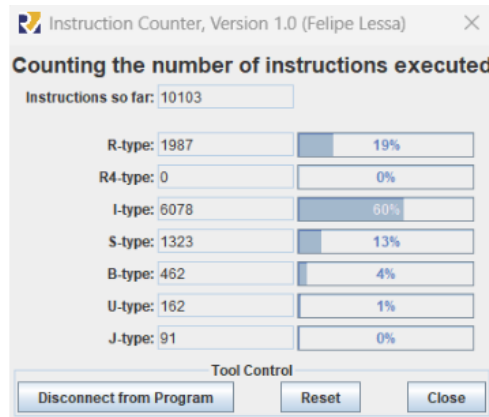
122      addi    a5,a5,-1
123      sw      a5,-24(s0)
124      j       L7
125 L9:
126      lw      a1,-24(s0)
127      lw      a0,-36(s0)
128      jal     swap          #call mudado para jal
129      lw      a5,-24(s0)
130      addi    a5,a5,-1
131      sw      a5,-24(s0)
132 L7:
133      lw      a5,-24(s0)
134      blt     a5,zero,L8
135      lw      a5,-24(s0)
136      slli    a5,a5,2
137      lw      a4,-36(s0)
138      add     a5,a4,a5
139      lw      a4,0(a5)
140      lw      a5,-24(s0)
141      addi    a5,a5,1
142      slli    a5,a5,2
143      lw      a3,-36(s0)
144      add     a5,a3,a5
145      lw      a5,0(a5)
146      bgt     a4,a5,L9
147 L8:
148      lw      a5,-20(s0)
149      addi    a5,a5,1
150      sw      a5,-20(s0)
151 L6:
152      lw      a4,-20(s0)
153      lw      a5,-40(s0)
154      blt     a4,a5,L10
155      #nop
156      #nop
157      lw      ra,44(sp)
158      lw      s0,40(sp)
159      addi    sp,sp,48
160      jr      ra

```

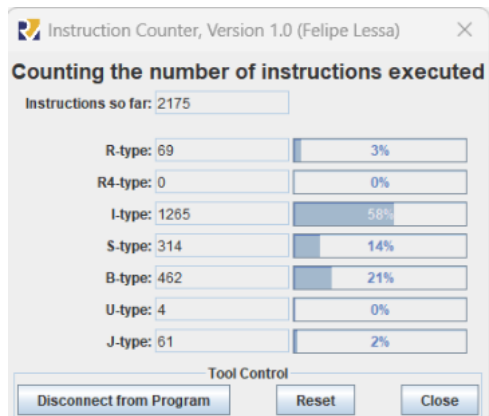
2.3)

É válido notar que o Compile Explorer fornece um código com uma diretiva `.set` que não existe no RARS. Desse modo, algumas mudanças foram feitas, especificamente na posição da função `main` e no carregamento do endereço do vetor.

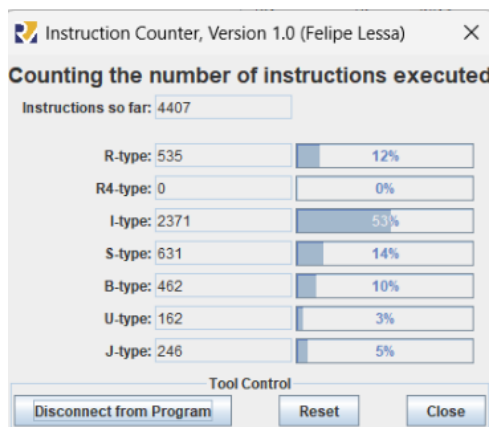
Para a otimização `-O0`, o total de instruções é de 10103 instruções



Para a otimização `-O3`, o total de instruções é 2175



Para otimização `-Os`, o total de instruções é 4407



	instruções	tamanho em bytes
-O0	10103	540
-O3	2175	380
-Os	4407	336
-sort ex 1.1	3746	300

2.4)

As opções -O0, -O1, -O2, -O3 e -Os são níveis de otimização do compilador e controlam o quanto o compilador tenta melhorar o desempenho e reduzir o tamanho do código gerado.

No nível -O0, praticamente não há otimização. O compilador gera um código muito próximo do que foi escrito no código-fonte, com instruções simples e diretas. Isso facilita o debug, porque cada linha de código se alinha diretamente com uma ou poucas instruções de máquina. Mas, o programa costuma rodar de forma mais lenta e ocupar mais espaço, já que não há tentativa de eliminar redundâncias nem reorganizar instruções.

Com -O1, o compilador começa a aplicar otimizações básicas, como remoção de trechos do código que nunca são executados, simplificação de expressões e eliminação de variáveis temporárias desnecessárias. O código final fica um pouco mais eficiente, mas ainda preserva uma estrutura que facilita a depuração.

Já no nível -O2, o compilador passa a aplicar otimizações mais agressivas, incluindo melhor uso de registradores, reordenação de instruções para reduzir esperas e maior remoção de redundâncias. Essa opção busca um bom equilíbrio entre velocidade e tamanho do binário e por isso é a otimização mais usada para programas de produção, já que oferece ganhos significativos de desempenho sem aumentar muito o tempo de compilação.

Com -O3, o compilador leva as otimizações do -O2 mais a fundo, adicionando ainda mais estratégias focadas em desempenho máximo, como unrolling de loops (desenrolar laços para reduzir saltos) e vetorização automática. Isso pode gerar código bem mais rápido, mas também mais difícil de analisar e depurar, além de aumentar o tamanho do código e o tempo de compilação.

Por fim, -Os tem um objetivo diferente: foca em otimizar para tamanho do programa, em vez de apenas desempenho. Ele aplica muitas otimizações do -O2, mas evita transformações que aumentariam significativamente o tamanho do código. É muito usado em sistemas embarcados ou ambientes com memória limitada.

0.3 Item 3

Transformada Discreta de Fourier

3.1)

Primeiramente, dois códigos em C foram desenvolvidos com o fito de verificar a eficácia computacional da aproximação trigonométrica por meio de séries. Após esse processo, os programas foram compilados no Compile Explorer (RISC-V (32bits) gcc 15.2.0) afim de se obter o arquivo .s de ambos os códigos com otimização -O0 e rodá-los no RARS. Após mudanças similares às mencionadas no item 2.2, cada um funcionou individualmente de forma correta. Por fim, ambos os códigos em C foram mesclados, compilado sobre as mesmas condições mencionadas anteriormente e obteve-se um programa que calcula a aproximação do $\sin()$ e $\cos()$ que é executado no RARS. Notou-se uma margem de erro na aproximação a partir de um certo N, de modo que independente de quantas somas fossem feitas, o erro se mantinha o mesmo. Isso se deve ao erro inerente ao uso de floats em sistemas computacionais. Para tal, foram usadas as aproximações por séries:

Calcula Cosseno em C:

```
1 float calccosseno(float radianos, int termos) {
2     const float EPS = 1e-7f;
3     float x = normalize_angle(radianos);
4
5     if (x < 0.0f) x = -x;
6
7     /* reduzir para [0, PI/2] usando cos(pi - t) = -cos(t) */
8     int sign = 1;
9     if (x > (PI * 0.5f)) {
10         x = PI - x;
11         sign = -1;
12     }
13
14     float x2 = x * x;
15     float term = 1.0f;      /* k=0 */
16     float sum = term;
17
18     for (int k = 1; k < termos; ++k) {
19         float denom = (2.0f * k - 1.0f) * (2.0f * k);
20         term *= - x2 / denom; /* termo recursivo */
21         sum += term;
22         if (term < 0.0f ? -term < EPS : term < EPS) break;
```

```

23     }
24     return sign * sum;
25 }

```

Cálcula Seno:

```

1 float calcseno(float radianos, int termos) {
2     const float EPS = 1e-7f;
3     float x = normalize_angle(radianos);
4
5     int sign = 1;
6     if (x < 0.0f) { x = -x; sign = -1; }
7
8     /* reduzir para [0, PI/2] usando sin(pi - t) = sin(t) */
9     if (x > (PI * 0.5f)) {
10         x = PI - x;
11     }
12
13     float x2 = x * x;
14     float term = x;      /* k=0 */
15     float sum = term;
16
17     for (int k = 1; k < termos; ++k) {
18         float denom = (2.0f * k) * (2.0f * k + 1.0f);
19         term *= - x2 / denom;
20         sum += term;
21         if (term < 0.0f ? -term < EPS : term < EPS) break;
22     }
23     return sign * sum;
24 }

```

Em Assembly essas funções estavam funcionando perfeitamente, mas apenas para valores entre 0 e 2π , ou seja, uma nova função deveria ser criada a fim de normalizar esses ângulos para funcionar conforme o esperado:

```

1 float normalize_angle(float a) {
2     float two_pi = 2.0f * PI;
3     /* reduz uma vez por divis o inteira (trunca zero) */
4     int k = (int)(a / two_pi);
5     a -= (float)k * two_pi;
6     /* ajusta (-PI, PI] */
7     if (a <= -PI) a += two_pi;
8     if (a > PI) a -= two_pi;

```

```

9     return a;
10 }

```

Nesse sentido, o código em Assembly de todas essas funções juntas, compiladas em -O0 é:

```

1  addi    sp,sp,-48
2          sw      ra,44(sp)
3          sw      s0,40(sp)
4          addi    s0,sp,48
5          fsw     fa0,-36(s0)
6          lui     a5,%hi(.LC0)
7          flw     fa5,%lo(.LC0)(a5)
8          fsw     fa5,-20(s0)
9          flw     fa4,-36(s0)
10         flw     fa5,-20(s0)
11         fdiv.s  fa5,fa4,fa5
12         fcvt.w.s a5,fa5,rtz
13         sw      a5,-24(s0)
14         lw      a5,-24(s0)
15         fcvt.s.w      fa4,a5
16         flw     fa5,-20(s0)
17         fmul.s  fa5,fa4,fa5
18         flw     fa4,-36(s0)
19         fsub.s  fa5,fa4,fa5
20         fsw     fa5,-36(s0)
21         flw     fa4,-36(s0)
22         lui     a5,%hi(.LC1)
23         flw     fa5,%lo(.LC1)(a5)
24         fle.s   a5,fa4,fa5
25         bne     a5,zero,.L13
26         j       .L8
27 .L13:
28         flw     fa4,-36(s0)
29         flw     fa5,-20(s0)
30         fadd.s  fa5,fa4,fa5
31         fsw     fa5,-36(s0)
32 .L8:
33         flw     fa4,-36(s0)
34         lui     a5,%hi(.LC2)
35         flw     fa5,%lo(.LC2)(a5)

```

```

36      fgt.s    a5,fa4,fa5
37      bne     a5,zero,.L14
38      j       .L10
39 .L14:
40      flw     fa4,-36(s0)
41      flw     fa5,-20(s0)
42      fsub.s   fa5,fa4,fa5
43      fsw     fa5,-36(s0)
44 .L10:
45      flw     fa5,-36(s0)
46      fmv.s    fa0,fa5
47      lw      ra,44(sp)
48      lw      s0,40(sp)
49      addi    sp,sp,48
50      jr      ra
51 calccosseno:
52      addi    sp,sp,-64
53      sw      ra,60(sp)
54      sw      s0,56(sp)
55      addi    s0,sp,64
56      fsw     fa0,-52(s0)
57      sw      a0,-56(s0)
58      lui     a5,%hi(.LC3)
59      flw     fa5,%lo(.LC3)(a5)
60      fsw     fa5,-40(s0)
61      flw     fa0,-52(s0)
62      jal     normalize_angle
63      fsw     fa0,-20(s0)
64      flw     fa5,-20(s0)
65      fmv.s.x fa4,zero
66      flt.s    a5,fa5,fa4
67      bne     a5,zero,.L28
68      j       .L16
69 .L28:
70      flw     fa5,-20(s0)
71      fneg.s   fa5,fa5
72      fsw     fa5,-20(s0)
73 .L16:
74      li      a5,1
75      sw      a5,-24(s0)
76      flw     fa4,-20(s0)

```

```

77      lui      a5,%hi(.LC4)
78      flw      fa5,%lo(.LC4)(a5)
79      fgt.s    a5,fa4,fa5
80      bne      a5,zero,.L29
81      j        .L18
82  .L29:
83      lui      a5,%hi(.LC2)
84      flw      fa4,%lo(.LC2)(a5)
85      flw      fa5,-20(s0)
86      fsub.s   fa5,fa4,fa5
87      fsw      fa5,-20(s0)
88      li       a5,-1
89      sw       a5,-24(s0)
90  .L18:
91      flw      fa5,-20(s0)
92      fmul.s   fa5,fa5,fa5
93      fsw      fa5,-44(s0)
94      lui      a5,%hi(.LC5)
95      flw      fa5,%lo(.LC5)(a5)
96      fsw      fa5,-28(s0)
97      flw      fa5,-28(s0)
98      fsw      fa5,-32(s0)
99      li       a5,1
100     sw       a5,-36(s0)
101     j        .L20
102  .L26:
103     lw       a5,-36(s0)
104     fcvt.s.w          fa5,a5
105     fadd.s   fa4,fa5,fa5
106     lui      a5,%hi(.LC5)
107     flw      fa5,%lo(.LC5)(a5)
108     fsub.s   fa4,fa4,fa5
109     lw       a5,-36(s0)
110     fcvt.s.w          fa5,a5
111     fadd.s   fa5,fa5,fa5
112     fmul.s   fa5,fa4,fa5
113     fsw      fa5,-48(s0)
114     flw      fa5,-44(s0)
115     fneg.s   fa4,fa5
116     flw      fa5,-48(s0)
117     fdiv.s   fa5,fa4,fa5

```

```

118         flw      fa4, -28(s0)
119         fmul.s   fa5, fa4, fa5
120         fsw      fa5, -28(s0)
121         flw      fa4, -32(s0)
122         flw      fa5, -28(s0)
123         fadd.s   fa5, fa4, fa5
124         fsw      fa5, -32(s0)
125         flw      fa5, -28(s0)
126         fmv.s.x  fa4, zero
127         flt.s    a5, fa5, fa4
128         bne      a5, zero, .L30
129         j        .L31
130 .L30:
131         flw      fa5, -28(s0)
132         fneg.s   fa5, fa5
133         flw      fa4, -40(s0)
134         fgt.s    a5, fa4, fa5
135         andi     a5, a5, 0xff
136         j        .L23
137 .L31:
138         flw      fa4, -28(s0)
139         flw      fa5, -40(s0)
140         flt.s    a5, fa4, fa5
141         andi     a5, a5, 0xff
142 .L23:
143         bne      a5, zero, .L32
144         lw       a5, -36(s0)
145         addi     a5, a5, 1
146         sw       a5, -36(s0)
147 .L20:
148         lw       a4, -36(s0)
149         lw       a5, -56(s0)
150         blt      a4, a5, .L26
151         j        .L25
152 .L32:
153         nop
154 .L25:
155         lw       a5, -24(s0)
156         fcvt.s.w          fa4, a5
157         flw      fa5, -32(s0)
158         fmul.s   fa5, fa4, fa5

```

```

159      fmv.s    fa0,fa5
160      lw      ra,60(sp)
161      lw      s0,56(sp)
162      addi    sp,sp,64
163      jr      ra
164 calcseno:
165      addi    sp,sp,-64
166      sw      ra,60(sp)
167      sw      s0,56(sp)
168      addi    s0,sp,64
169      fsw     fa0,-52(s0)
170      sw      a0,-56(s0)
171      lui     a5,%hi(.LC3)
172      flw     fa5,%lo(.LC3)(a5)
173      fsw     fa5,-40(s0)
174      flw     fa0,-52(s0)
175      jal     normalize_angle
176      fsw     fa0,-20(s0)
177      li      a5,1
178      sw      a5,-24(s0)
179      flw     fa5,-20(s0)
180      fmv.s.x fa4,zero
181      flt.s   a5,fa5,fa4
182      bne     a5,zero,.L46
183      j       .L34
184 .L46:
185      flw     fa5,-20(s0)
186      fneg.s  fa5,fa5
187      fsw     fa5,-20(s0)
188      li      a5,-1
189      sw      a5,-24(s0)
190 .L34:
191      flw     fa4,-20(s0)
192      lui     a5,%hi(.LC4)
193      flw     fa5,%lo(.LC4)(a5)
194      fgt.s   a5,fa4,fa5
195      bne     a5,zero,.L47
196      j       .L36
197 .L47:
198      lui     a5,%hi(.LC2)
199      flw     fa4,%lo(.LC2)(a5)

```

```

200         flw      fa5,-20(s0)
201         fsub.s   fa5,fa4,fa5
202         fsw      fa5,-20(s0)
203 .L36:
204         flw      fa5,-20(s0)
205         fmul.s   fa5,fa5,fa5
206         fsw      fa5,-44(s0)
207         flw      fa5,-20(s0)
208         fsw      fa5,-28(s0)
209         flw      fa5,-28(s0)
210         fsw      fa5,-32(s0)
211         li       a5,1
212         sw       a5,-36(s0)
213         j        .L38
214 .L44:
215         lw       a5,-36(s0)
216         fcvt.s.w          fa5,a5
217         fadd.s   fa4,fa5,fa5
218         lw       a5,-36(s0)
219         fcvt.s.w          fa5,a5
220         fadd.s   fa3,fa5,fa5
221         lui      a5,%hi(.LC5)
222         flw      fa5,%lo(.LC5)(a5)
223         fadd.s   fa5,fa3,fa5
224         fmul.s   fa5,fa4,fa5
225         fsw      fa5,-48(s0)
226         flw      fa5,-44(s0)
227         fneg.s   fa4,fa5
228         flw      fa5,-48(s0)
229         fdiv.s   fa5,fa4,fa5
230         flw      fa4,-28(s0)
231         fmul.s   fa5,fa4,fa5
232         fsw      fa5,-28(s0)
233         flw      fa4,-32(s0)
234         flw      fa5,-28(s0)
235         fadd.s   fa5,fa4,fa5
236         fsw      fa5,-32(s0)
237         flw      fa5,-28(s0)
238         fmv.s.x  fa4,zero
239         flt.s    a5,fa5,fa4
240         bne      a5,zero,.L48

```



```

241         j            .L49
242 .L48:
243         flw          fa5,-28(s0)
244         fneg.s       fa5,fa5
245         flw          fa4,-40(s0)
246         fgt.s        a5,fa4,fa5
247         andi         a5,a5,0xff
248         j            .L41
249 .L49:
250         flw          fa4,-28(s0)
251         flw          fa5,-40(s0)
252         flt.s        a5,fa4,fa5
253         andi         a5,a5,0xff
254 .L41:
255         bne          a5,zero,.L50
256         lw           a5,-36(s0)
257         addi         a5,a5,1
258         sw           a5,-36(s0)
259 .L38:
260         lw           a4,-36(s0)
261         lw           a5,-56(s0)
262         blt          a4,a5,.L44
263         j            .L43
264 .L50:
265         nop
266 .L43:
267         lw           a5,-24(s0)
268         fcvt.s.w      fa4,a5
269         flw          fa5,-32(s0)
270         fmul.s       fa5,fa4,fa5
271         fmv.s        fa0,fa5
272         lw           ra,60(sp)
273         lw           s0,56(sp)
274         addi         sp,sp,64
275         jr           ra

```

Funcionamento direto na função completa de DFT:

```
sin = 0.0
cos = 1.0
sin = 0.0
cos = 1.0
sin = 0.0
cos = 0.70710677
sin = -0.70710677
cos = -6.25385E-8
sin = -0.99999994
cos = -0.70710677
```

3.2)

O procedimento da Transformada Discreta de Fourier foi escrito inicialmente em C e após verificada a precisão dos resultados da função em C, usou-se a função obtida no item 3.1 para substituir o valor de e elevado a $(i \cdot \theta)$ por $\cos(\theta) + i \cdot \sin(\theta)$, conforme sugerido na questão. Posteriormente, o código completo foi compilado em -O3 utilizando a ferramenta Compile Explorer RISC-V (32bits) gcc 15.2.0.

Código da Transformada Discreta de Fourier escrito em C, onde as funções `cosf` e `sinf` foram posteriormente substituídas pelo código do item 3.1:

```
1 void DFT(float *x, float *X_real, float *X_imag, int N) {
2     for (int k = 0; k < N; k++) {
3         float soma_real = 0.0f;
4         float soma_imag = 0.0f;
5
6         for (int n = 0; n < N; n++) {
7             float ang = -2.0f * PI * k * n / N;
8             soma_real += x[n] * cosf(ang);
9             soma_imag += x[n] * sinf(ang);
10        }
11
12        X_real[k] = soma_real;
13        X_imag[k] = soma_imag;
14    }
15 }
```

Procedimento completo da Transformada Discreta de Fourier (Incluindo o código do somatório unido ao código para calcular seno e cosseno):

```

1  .data
2
3  newline: .string "\n"
4  space:   .string " + i*"
5  N:       .word    8
6  x:       .float   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
7  X_real:  .space   32
8  X_imag:  .space   32
9
10 .LC0:
11     .word    1086918619##2pi
12 .LC1:
13     .word    -1068953637##pi
14 .LC2:
15     .word    1078530011#pi tbm?
16 .LC3:
17     .word    869711765##10-7
18 .LC4:
19     .word    1070141403##pi/2
20 .LC5:
21     .word    1065353216## primeiro termo dos somatorios
22 .LC6:
23     .word    -1060565029 ## -2pi
24
25
26 .text
27
28 main:
29     # --- Pr logo ---
30     addi     sp, sp, -32
31     sw       ra, 28(sp)
32     sw       s0, 24(sp)
33     sw       s1, 20(sp)
34     sw       s2, 16(sp)
35     sw       s3, 12(sp)
36     csrr     s0, 3073
37     la       a0, x
38     la       a1, X_real

```

```

39     la      a2, X_imag
40     lw      a3, N
41     jal     DFT
42     csrr    t0, 3073
43     sub     s0, t0, s0
44     # mv     a1, s0
45     # li     a0, 1
46     # ecall
47     # la     a1, newline
48     # li     a0, 4
49     # ecall
50     li      s1, 0
51     la      s2, X_real
52     la      s3, X_imag
53     lw      a5, N
54 print_loop_start:
55     bge     s1, a5, print_loop_end
56     slli    t0, s1, 2
57     add     t1, s2, t0
58     flw     fa0, 0(t1)
59     li      a7, 2
60     ecall
61     la      a0, space
62     li      a7, 4
63     ecall
64     add     t2, s3, t0
65     flw     fa0, 0(t2)
66     li      a7, 2
67     ecall
68     la      a0, newline
69     li      a7, 4
70     ecall
71     addi    s1, s1, 1
72     j       print_loop_start
73 print_loop_end:
74     li      a0, 0
75     lw      s3, 12(sp)
76     lw      s2, 16(sp)
77     lw      s1, 20(sp)
78     lw      s0, 24(sp)
79     lw      ra, 28(sp)

```

```

80      li      a7, 10
81      ecall
82      addi    sp, sp, 32
83      jr      ra
84 normalize_angle:
85          addi    sp, sp, -48
86          sw      ra, 44(sp)
87          sw      s0, 40(sp)
88          addi    s0, sp, 48
89          fsw     fa0, -36(s0)
90          lui     a5, %hi(.LC0)
91          flw     fa5, %lo(.LC0)(a5)
92          fsw     fa5, -20(s0)
93          flw     fa4, -36(s0)
94          flw     fa5, -20(s0)
95          fdiv.s  fa5, fa4, fa5
96          fcvt.w.s a5, fa5, rtz
97          sw      a5, -24(s0)
98          lw      a5, -24(s0)
99          fcvt.s.w      fa4, a5
100         flw     fa5, -20(s0)
101         fmul.s  fa5, fa4, fa5
102         flw     fa4, -36(s0)
103         fsub.s  fa5, fa4, fa5
104         fsw     fa5, -36(s0)
105         flw     fa4, -36(s0)
106         lui     a5, %hi(.LC1)
107         flw     fa5, %lo(.LC1)(a5)
108         fle.s   a5, fa4, fa5
109         bne     a5, zero, .L13
110         j       .L8
111 .L13:
112         flw     fa4, -36(s0)
113         flw     fa5, -20(s0)
114         fadd.s  fa5, fa4, fa5
115         fsw     fa5, -36(s0)
116 .L8:
117         flw     fa4, -36(s0)
118         lui     a5, %hi(.LC2)
119         flw     fa5, %lo(.LC2)(a5)
120         fgt.s   a5, fa4, fa5

```

```

121         bne      a5,zero,.L14
122         j        .L10
123 .L14:
124         flw      fa4,-36(s0)
125         flw      fa5,-20(s0)
126         fsub.s   fa5,fa4,fa5
127         fsw      fa5,-36(s0)
128 .L10:
129         flw      fa5,-36(s0)
130         fmv.s    fa0,fa5
131         lw       ra,44(sp)
132         lw       s0,40(sp)
133         addi     sp,sp,48
134         jr       ra
135 calccosseno:
136         addi     sp,sp,-64
137         sw       ra,60(sp)
138         sw       s0,56(sp)
139         addi     s0,sp,64
140         fsw      fa0,-52(s0)
141         sw       a0,-56(s0)
142         lui      a5,%hi(.LC3)
143         flw      fa5,%lo(.LC3)(a5)
144         fsw      fa5,-40(s0)
145         flw      fa0,-52(s0)
146         jal      normalize_angle
147         fsw      fa0,-20(s0)
148         flw      fa5,-20(s0)
149         fmv.s.x  fa4,zero
150         flt.s    a5,fa5,fa4
151         bne      a5,zero,.L28
152         j        .L16
153 .L28:
154         flw      fa5,-20(s0)
155         fneg.s   fa5,fa5
156         fsw      fa5,-20(s0)
157 .L16:
158         li       a5,1
159         sw       a5,-24(s0)
160         flw      fa4,-20(s0)
161         lui      a5,%hi(.LC4)

```

```

162         flw      fa5,%lo(.LC4)(a5)
163         fgt.s    a5,fa4,fa5
164         bne     a5,zero,.L29
165         j       .L18
166 .L29:
167         lui     a5,%hi(.LC2)
168         flw     fa4,%lo(.LC2)(a5)
169         flw     fa5,-20(s0)
170         fsub.s   fa5,fa4,fa5
171         fsw     fa5,-20(s0)
172         li      a5,-1
173         sw      a5,-24(s0)
174 .L18:
175         flw     fa5,-20(s0)
176         fmul.s   fa5,fa5,fa5
177         fsw     fa5,-44(s0)
178         lui     a5,%hi(.LC5)
179         flw     fa5,%lo(.LC5)(a5)
180         fsw     fa5,-28(s0)
181         flw     fa5,-28(s0)
182         fsw     fa5,-32(s0)
183         li      a5,1
184         sw      a5,-36(s0)
185         j       .L20
186 .L26:
187         lw      a5,-36(s0)
188         fcvt.s.w      fa5,a5
189         fadd.s   fa4,fa5,fa5
190         lui     a5,%hi(.LC5)
191         flw     fa5,%lo(.LC5)(a5)
192         fsub.s   fa4,fa4,fa5
193         lw      a5,-36(s0)
194         fcvt.s.w      fa5,a5
195         fadd.s   fa5,fa5,fa5
196         fmul.s   fa5,fa4,fa5
197         fsw     fa5,-48(s0)
198         flw     fa5,-44(s0)
199         fneg.s   fa4,fa5
200         flw     fa5,-48(s0)
201         fdiv.s   fa5,fa4,fa5
202         flw     fa4,-28(s0)

```

```

203      fmul.s   fa5,fa4,fa5
204      fsw      fa5,-28(s0)
205      flw      fa4,-32(s0)
206      flw      fa5,-28(s0)
207      fadd.s   fa5,fa4,fa5
208      fsw      fa5,-32(s0)
209      flw      fa5,-28(s0)
210      fmv.s.x  fa4,zero
211      flt.s    a5,fa5,fa4
212      bne      a5,zero,.L30
213      j        .L31
214 .L30:
215      flw      fa5,-28(s0)
216      fneg.s   fa5,fa5
217      flw      fa4,-40(s0)
218      fgt.s    a5,fa4,fa5
219      andi     a5,a5,0xff
220      j        .L23
221 .L31:
222      flw      fa4,-28(s0)
223      flw      fa5,-40(s0)
224      flt.s    a5,fa4,fa5
225      andi     a5,a5,0xff
226 .L23:
227      bne      a5,zero,.L32
228      lw       a5,-36(s0)
229      addi     a5,a5,1
230      sw       a5,-36(s0)
231 .L20:
232      lw       a4,-36(s0)
233      lw       a5,-56(s0)
234      blt      a4,a5,.L26
235      j        .L25
236 .L32:
237      nop
238 .L25:
239      lw       a5,-24(s0)
240      fcvt.s.w          fa4,a5
241      flw      fa5,-32(s0)
242      fmul.s   fa5,fa4,fa5
243      fmv.s    fa0,fa5

```



```

244         lw      ra,60(sp)
245         lw      s0,56(sp)
246         addi    sp,sp,64
247         jr      ra
248 calcseno:
249         addi    sp,sp,-64
250         sw      ra,60(sp)
251         sw      s0,56(sp)
252         addi    s0,sp,64
253         fsw     fa0,-52(s0)
254         sw      a0,-56(s0)
255         lui     a5,%hi(.LC3)
256         flw     fa5,%lo(.LC3)(a5)
257         fsw     fa5,-40(s0)
258         flw     fa0,-52(s0)
259         jal     normalize_angle
260         fsw     fa0,-20(s0)
261         li      a5,1
262         sw      a5,-24(s0)
263         flw     fa5,-20(s0)
264         fmv.s.x fa4,zero
265         flt.s   a5,fa5,fa4
266         bne     a5,zero,.L46
267         j       .L34
268 .L46:
269         flw     fa5,-20(s0)
270         fneg.s  fa5,fa5
271         fsw     fa5,-20(s0)
272         li      a5,-1
273         sw      a5,-24(s0)
274 .L34:
275         flw     fa4,-20(s0)
276         lui     a5,%hi(.LC4)
277         flw     fa5,%lo(.LC4)(a5)
278         fgt.s   a5,fa4,fa5
279         bne     a5,zero,.L47
280         j       .L36
281 .L47:
282         lui     a5,%hi(.LC2)
283         flw     fa4,%lo(.LC2)(a5)
284         flw     fa5,-20(s0)

```

```

285      fsub.s   fa5,fa4,fa5
286      fsw     fa5,-20(s0)
287  .L36:
288      flw     fa5,-20(s0)
289      fmul.s   fa5,fa5,fa5
290      fsw     fa5,-44(s0)
291      flw     fa5,-20(s0)
292      fsw     fa5,-28(s0)
293      flw     fa5,-28(s0)
294      fsw     fa5,-32(s0)
295      li      a5,1
296      sw      a5,-36(s0)
297      j       .L38
298  .L44:
299      lw      a5,-36(s0)
300      fcvt.s.w      fa5,a5
301      fadd.s   fa4,fa5,fa5
302      lw      a5,-36(s0)
303      fcvt.s.w      fa5,a5
304      fadd.s   fa3,fa5,fa5
305      lui     a5,%hi(.LC5)
306      flw     fa5,%lo(.LC5)(a5)
307      fadd.s   fa5,fa3,fa5
308      fmul.s   fa5,fa4,fa5
309      fsw     fa5,-48(s0)
310      flw     fa5,-44(s0)
311      fneg.s   fa4,fa5
312      flw     fa5,-48(s0)
313      fdiv.s   fa5,fa4,fa5
314      flw     fa4,-28(s0)
315      fmul.s   fa5,fa4,fa5
316      fsw     fa5,-28(s0)
317      flw     fa4,-32(s0)
318      flw     fa5,-28(s0)
319      fadd.s   fa5,fa4,fa5
320      fsw     fa5,-32(s0)
321      flw     fa5,-28(s0)
322      fmv.s.x  fa4,zero
323      flt.s    a5,fa5,fa4
324      bne     a5,zero,.L48
325      j       .L49

```

```

326 .L48:
327     flw      fa5, -28(s0)
328     fneg.s   fa5, fa5
329     flw      fa4, -40(s0)
330     fgt.s    a5, fa4, fa5
331     andi     a5, a5, 0xff
332     j        .L41
333 .L49:
334     flw      fa4, -28(s0)
335     flw      fa5, -40(s0)
336     flt.s    a5, fa4, fa5
337     andi     a5, a5, 0xff
338 .L41:
339     bne      a5, zero, .L50
340     lw       a5, -36(s0)
341     addi     a5, a5, 1
342     sw       a5, -36(s0)
343 .L38:
344     lw       a4, -36(s0)
345     lw       a5, -56(s0)
346     blt      a4, a5, .L44
347     j        .L43
348 .L50:
349     nop
350 .L43:
351     lw       a5, -24(s0)
352     fcvts.w   fa4, a5
353     flw      fa5, -32(s0)
354     fmul.s    fa5, fa4, fa5
355     fmv.s     fa0, fa5
356     lw       ra, 60(sp)
357     lw       s0, 56(sp)
358     addi     sp, sp, 64
359     jr       ra
360 DFT:
361     addi     sp, sp, -64
362     sw       ra, 60(sp)
363     sw       s0, 56(sp)
364     addi     s0, sp, 64
365     sw       a0, -52(s0)
366     sw       a1, -56(s0)

```

```

367      sw      a2,-60(s0)
368      sw      a3,-64(s0)
369      li      a5,12
370      sw      a5,-36(s0)
371      sw      zero,-20(s0)
372      j       .L52
373 .L55:
374      sw      zero,-24(s0)
375      sw      zero,-28(s0)
376      sw      zero,-32(s0)
377      j       .L53
378 .L54:
379      lw      a5,-20(s0)
380      fcvt.s.w      fa4,a5
381      lui     a5,%hi(.LC6)
382      flw     fa5,%lo(.LC6)(a5)
383      fmul.s  fa4,fa4,fa5
384      lw      a5,-32(s0)
385      fcvt.s.w      fa5,a5
386      fmul.s  fa4,fa4,fa5
387      lw      a5,-64(s0)
388      fcvt.s.w      fa5,a5
389      fdiv.s  fa5,fa4,fa5
390      fsw     fa5,-40(s0)
391      lw      a0,-36(s0)
392      flw     fa0,-40(s0)
393      jal     calccoseno
394      fsw     fa0,-44(s0)
395      lw      a0,-36(s0)
396      flw     fa0,-40(s0)
397      jal     calcseno
398      fsw     fa0,-48(s0)
399      lw      a5,-32(s0)
400      slli    a5,a5,2
401      lw      a4,-52(s0)
402      add     a5,a4,a5
403      flw     fa4,0(a5)
404      flw     fa5,-44(s0)
405      fmul.s  fa5,fa4,fa5
406      flw     fa4,-24(s0)
407      fadd.s  fa5,fa4,fa5

```

```

408      fsw      fa5, -24(s0)
409      lw       a5, -32(s0)
410      slli     a5, a5, 2
411      lw       a4, -52(s0)
412      add      a5, a4, a5
413      flw      fa4, 0(a5)
414      flw      fa5, -48(s0)
415      fmul.s   fa5, fa4, fa5
416      flw      fa4, -28(s0)
417      fadd.s   fa5, fa4, fa5
418      fsw      fa5, -28(s0)
419      lw       a5, -32(s0)
420      addi     a5, a5, 1
421      sw       a5, -32(s0)
422 .L53:
423      lw       a4, -32(s0)
424      lw       a5, -64(s0)
425      blt      a4, a5, .L54
426      ## lw     a5, -20(s0)
427      ## slli   a5, a5, 2
428      ## lw     a4, -56(s0)
429      ## add    a5, a4, a5
430      ## flw    fa5, -24(s0)
431      ## fsw    fa5, 0(a5)
432      #lw      a5, -20(s0)
433      #slli    a5, a5, 2
434      #lw      a4, -60(s0)
435      #add     a5, a4, a5
436      #flw     fa5, -28(s0)
437      #fsw     fa5, 0(a5)
438      lw      t0, -20(s0)
439      slli     t1, t0, 2
440      lw      a4, -56(s0)
441      add     a5, a4, t1
442      flw     fa5, -24(s0)
443      fsw     fa5, 0(a5)
444      lw      a4, -60(s0)
445      add     a5, a4, t1
446      flw     fa5, -28(s0)
447      fsw     fa5, 0(a5)
448      lw      a5, -20(s0)

```

```

449         addi    a5,a5,1
450         sw      a5,-20(s0)
451 .L52:
452         lw      a4,-20(s0)
453         lw      a5,-64(s0)
454         blt     a4,a5,.L55
455         nop
456         nop
457         lw      ra,60(sp)
458         lw      s0,56(sp)
459         addi    sp,sp,64
460         jr      ra

```

3.3)

Utilizando o código obtido no item 3.2, define-se os 8 valores do vetor $x[k]$ como floats de valor 1.0:

```

1  .data
2
3  newline: .string "\n"
4  space:   .string " + i*"
5
6  N:       .word    8
7  x:       .float   1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
8  X_real:  .space   32
9  X_imag:  .space   32
10 cos_label: .string "cos = "
11 sin_label: .string "sin = "
12
13 .LC0:
14         .word    1086918619##2pi
15 .LC1:
16         .word    -1068953637##pi
17 .LC2:
18         .word    1078530011#pi tbm?
19 .LC3:
20         .word    869711765##10-7
21 .LC4:
22         .word    1070141403##pi/2
23 .LC5:
24         .word    1065353216## primeiro termo dos somatorios

```

```

25 .LC6:
26         .word    -1060565029 ## -2pi
27
28
29 .text
30
31 main:
32     addi    sp, sp, -32
33     sw      ra, 28(sp)
34     sw      s0, 24(sp)
35     sw      s1, 20(sp)
36     sw      s2, 16(sp)
37     sw      s3, 12(sp)
38     csrr    s0, 3073
39     la      a0, x
40     la      a1, X_real
41     la      a2, X_imag
42     lw      a3, N
43     jal     DFT
44     csrr    t0, 3073
45     sub     s0, t0, s0
46     mv      a1, s0
47     li      a7, 1
48     ecall
49     la      a1, newline
50     li      a7, 4
51     ecall
52     li      s1, 0
53     la      s2, X_real
54     la      s3, X_imag
55     lw      a5, N
56 print_loop_start:
57     bge     s1, a5, print_loop_end
58     slli    t0, s1, 2
59
60     add     t1, s2, t0
61     flw     fa0, 0(t1)
62     li      a7, 2
63     ecall
64
65

```

```

66     la      a0, space
67     li      a7, 4
68     ecalls
69     add     t2, s3, t0
70     flw     fa0, 0(t2)
71     li      a7, 2
72     ecalls
73     la      a0, newline
74     li      a7, 4
75     ecalls
76     addi    s1, s1, 1
77     j       print_loop_start
78
79 print_loop_end:
80     li      a0, 0
81     lw      s3, 12(sp)
82     lw      s2, 16(sp)
83     lw      s1, 20(sp)
84     lw      s0, 24(sp)
85     lw      ra, 28(sp)
86     li      a7, 10
87     ecalls
88     addi    sp, sp, 32
89     jr      ra
90 normalize_angle:
91     addi    sp, sp, -48
92     sw      ra, 44(sp)
93     sw      s0, 40(sp)
94     addi    s0, sp, 48
95     fsw     fa0, -36(s0)
96     lui     a5, %hi(.LC0)
97     flw     fa5, %lo(.LC0)(a5)
98     fsw     fa5, -20(s0)
99     flw     fa4, -36(s0)
100    flw     fa5, -20(s0)
101    fdiv.s   fa5, fa4, fa5
102    fcvt.w.s a5, fa5, rtz
103    sw      a5, -24(s0)
104    lw      a5, -24(s0)
105    fcvt.s.w          fa4, a5
106    flw     fa5, -20(s0)

```



```

107      fmul.s   fa5,fa4,fa5
108      flw      fa4,-36(s0)
109      fsub.s   fa5,fa4,fa5
110      fsw      fa5,-36(s0)
111      flw      fa4,-36(s0)
112      lui      a5,%hi(.LC1)
113      flw      fa5,%lo(.LC1)(a5)
114      fle.s    a5,fa4,fa5
115      bne      a5,zero,.L13
116      j        .L8
117 .L13:
118      flw      fa4,-36(s0)
119      flw      fa5,-20(s0)
120      fadd.s   fa5,fa4,fa5
121      fsw      fa5,-36(s0)
122 .L8:
123      flw      fa4,-36(s0)
124      lui      a5,%hi(.LC2)
125      flw      fa5,%lo(.LC2)(a5)
126      fgt.s    a5,fa4,fa5
127      bne      a5,zero,.L14
128      j        .L10
129 .L14:
130      flw      fa4,-36(s0)
131      flw      fa5,-20(s0)
132      fsub.s   fa5,fa4,fa5
133      fsw      fa5,-36(s0)
134 .L10:
135      flw      fa5,-36(s0)
136      fmv.s    fa0,fa5
137      lw       ra,44(sp)
138      lw       s0,40(sp)
139      addi     sp,sp,48
140      jr       ra
141 calccosseno:
142      addi     sp,sp,-64
143      sw       ra,60(sp)
144      sw       s0,56(sp)
145      addi     s0,sp,64
146      fsw      fa0,-52(s0)
147      sw       a0,-56(s0)

```

```

148      lui      a5,%hi(.LC3)
149      flw      fa5,%lo(.LC3)(a5)
150      fsw      fa5,-40(s0)
151      flw      fa0,-52(s0)
152      jal      normalize_angle
153      fsw      fa0,-20(s0)
154      flw      fa5,-20(s0)
155      fmv.s.x  fa4,zero
156      flt.s    a5,fa5,fa4
157      bne      a5,zero,.L28
158      j        .L16
159 .L28:
160      flw      fa5,-20(s0)
161      fneg.s   fa5,fa5
162      fsw      fa5,-20(s0)
163 .L16:
164      li       a5,1
165      sw       a5,-24(s0)
166      flw      fa4,-20(s0)
167      lui      a5,%hi(.LC4)
168      flw      fa5,%lo(.LC4)(a5)
169      fgt.s    a5,fa4,fa5
170      bne      a5,zero,.L29
171      j        .L18
172 .L29:
173      lui      a5,%hi(.LC2)
174      flw      fa4,%lo(.LC2)(a5)
175      flw      fa5,-20(s0)
176      fsub.s   fa5,fa4,fa5
177      fsw      fa5,-20(s0)
178      li       a5,-1
179      sw       a5,-24(s0)
180 .L18:
181      flw      fa5,-20(s0)
182      fmul.s   fa5,fa5,fa5
183      fsw      fa5,-44(s0)
184      lui      a5,%hi(.LC5)
185      flw      fa5,%lo(.LC5)(a5)
186      fsw      fa5,-28(s0)
187      flw      fa5,-28(s0)
188      fsw      fa5,-32(s0)

```

```

189      li      a5,1
190      sw      a5,-36(s0)
191      j       .L20
192 .L26:
193      lw      a5,-36(s0)
194      fcvt.s.w      fa5,a5
195      fadd.s  fa4,fa5,fa5
196      lui     a5,%hi(.LC5)
197      flw     fa5,%lo(.LC5)(a5)
198      fsub.s  fa4,fa4,fa5
199      lw      a5,-36(s0)
200      fcvt.s.w      fa5,a5
201      fadd.s  fa5,fa5,fa5
202      fmul.s  fa5,fa4,fa5
203      fsw     fa5,-48(s0)
204      flw     fa5,-44(s0)
205      fneg.s  fa4,fa5
206      flw     fa5,-48(s0)
207      fdiv.s  fa5,fa4,fa5
208      flw     fa4,-28(s0)
209      fmul.s  fa5,fa4,fa5
210      fsw     fa5,-28(s0)
211      flw     fa4,-32(s0)
212      flw     fa5,-28(s0)
213      fadd.s  fa5,fa4,fa5
214      fsw     fa5,-32(s0)
215      flw     fa5,-28(s0)
216      fmv.s.x fa4,zero
217      flt.s   a5,fa5,fa4
218      bne     a5,zero,.L30
219      j       .L31
220 .L30:
221      flw     fa5,-28(s0)
222      fneg.s  fa5,fa5
223      flw     fa4,-40(s0)
224      fgt.s   a5,fa4,fa5
225      andi    a5,a5,0xff
226      j       .L23
227 .L31:
228      flw     fa4,-28(s0)
229      flw     fa5,-40(s0)

```

```

230      flt.s    a5,fa4,fa5
231      andi     a5,a5,0xff
232  .L23:
233      bne      a5,zero,.L32
234      lw       a5,-36(s0)
235      addi     a5,a5,1
236      sw       a5,-36(s0)
237  .L20:
238      lw       a4,-36(s0)
239      lw       a5,-56(s0)
240      blt      a4,a5,.L26
241      j        .L25
242  .L32:
243      nop
244  .L25:
245      lw       a5,-24(s0)
246      fcvts.w  fa4,a5
247      flw      fa5,-32(s0)
248      fmul.s   fa5,fa4,fa5
249      fmv.s    fa0,fa5
250      lw       ra,60(sp)
251      lw       s0,56(sp)
252      addi     sp,sp,64
253      jr       ra
254  calcseno:
255      addi     sp,sp,-64
256      sw       ra,60(sp)
257      sw       s0,56(sp)
258      addi     s0,sp,64
259      fsw      fa0,-52(s0)
260      sw       a0,-56(s0)
261      lui      a5,%hi(.LC3)
262      flw      fa5,%lo(.LC3)(a5)
263      fsw      fa5,-40(s0)
264      flw      fa0,-52(s0)
265      jal      normalize_angle
266      fsw      fa0,-20(s0)
267      la       a0,newline
268      li       a5,1
269      sw       a5,-24(s0)
270      flw      fa5,-20(s0)

```

```

271      fmv.s.x fa4,zero
272      flt.s   a5,fa5,fa4
273      bne     a5,zero,.L46
274      j       .L34
275  .L46:
276      flw     fa5,-20(s0)
277      fneg.s  fa5,fa5
278      fsw     fa5,-20(s0)
279      li      a5,-1
280      sw      a5,-24(s0)
281  .L34:
282      flw     fa4,-20(s0)
283      lui     a5,%hi(.LC4)
284      flw     fa5,%lo(.LC4)(a5)
285      fgt.s   a5,fa4,fa5
286      bne     a5,zero,.L47
287      j       .L36
288  .L47:
289      lui     a5,%hi(.LC2)
290      flw     fa4,%lo(.LC2)(a5)
291      flw     fa5,-20(s0)
292      fsub.s  fa5,fa4,fa5
293      fsw     fa5,-20(s0)
294  .L36:
295      flw     fa5,-20(s0)
296      fmul.s  fa5,fa5,fa5
297      fsw     fa5,-44(s0)
298      flw     fa5,-20(s0)
299      fsw     fa5,-28(s0)
300      flw     fa5,-28(s0)
301      fsw     fa5,-32(s0)
302      li      a5,1
303      sw      a5,-36(s0)
304      j       .L38
305  .L44:
306      lw      a5,-36(s0)
307      fcvt.s.w      fa5,a5
308      fadd.s  fa4,fa5,fa5
309      lw      a5,-36(s0)
310      fcvt.s.w      fa5,a5
311      fadd.s  fa3,fa5,fa5

```

```

312      lui      a5,%hi(.LC5)
313      flw      fa5,%lo(.LC5)(a5)
314      fadd.s   fa5,fa3,fa5
315      fmul.s   fa5,fa4,fa5
316      fsw      fa5,-48(s0)
317      flw      fa5,-44(s0)
318      fneg.s   fa4,fa5
319      flw      fa5,-48(s0)
320      fdiv.s   fa5,fa4,fa5
321      flw      fa4,-28(s0)
322      fmul.s   fa5,fa4,fa5
323      fsw      fa5,-28(s0)
324      flw      fa4,-32(s0)
325      flw      fa5,-28(s0)
326      fadd.s   fa5,fa4,fa5
327      fsw      fa5,-32(s0)
328      flw      fa5,-28(s0)
329      fmv.s.x  fa4,zero
330      flt.s    a5,fa5,fa4
331      bne      a5,zero,.L48
332      j        .L49
333 .L48:
334      flw      fa5,-28(s0)
335      fneg.s   fa5,fa5
336      flw      fa4,-40(s0)
337      fgt.s    a5,fa4,fa5
338      andi     a5,a5,0xff
339      j        .L41
340 .L49:
341      flw      fa4,-28(s0)
342      flw      fa5,-40(s0)
343      flt.s    a5,fa4,fa5
344      andi     a5,a5,0xff
345 .L41:
346      bne      a5,zero,.L50
347      lw       a5,-36(s0)
348      addi     a5,a5,1
349      sw       a5,-36(s0)
350 .L38:
351      lw       a4,-36(s0)
352      lw       a5,-56(s0)

```

```

353      blt      a4,a5,.L44
354      j        .L43
355  .L50:
356      nop
357  .L43:
358      lw       a5,-24(s0)
359      fcvt.s.w      fa4,a5
360      flw      fa5,-32(s0)
361      fmul.s   fa5,fa4,fa5
362      fmv.s    fa0,fa5
363      lw       ra,60(sp)
364      lw       s0,56(sp)
365      addi     sp,sp,64
366      jr       ra
367  DFT:
368      addi     sp,sp,-64
369      sw       ra,60(sp)
370      sw       s0,56(sp)
371      addi     s0,sp,64
372      sw       a0,-52(s0)
373      sw       a1,-56(s0)
374      sw       a2,-60(s0)
375      sw       a3,-64(s0)
376      li      a5,12
377      sw       a5,-36(s0)
378      sw       zero,-20(s0)
379      j        .L52
380  .L55:
381      sw       zero,-24(s0)
382      sw       zero,-28(s0)
383      sw       zero,-32(s0)
384      j        .L53
385  .L54:
386      lw       a5,-20(s0)
387      fcvt.s.w      fa4,a5
388      lui      a5,%hi(.LC6)
389      flw      fa5,%lo(.LC6)(a5)
390      fmul.s   fa4,fa4,fa5
391      lw       a5,-32(s0)
392      fcvt.s.w      fa5,a5
393      fmul.s   fa4,fa4,fa5

```

```

394      lw      a5,-64(s0)
395      fcvt.s.w      fa5,a5
396      fdiv.s   fa5,fa4,fa5
397      fsw      fa5,-40(s0)
398      lw      a0,-36(s0)
399      flw      fa0,-40(s0)
400      jal      calccos seno
401      fsw      fa0,-44(s0)
402      lw      a0,-36(s0)
403      flw      fa0,-40(s0)
404      jal      calcseno
405      fsw      fa0,-48(s0)
406      lw      a5,-32(s0)
407      slli     a5,a5,2
408      lw      a4,-52(s0)
409      add      a5,a4,a5
410      flw      fa4,0(a5)
411      flw      fa5,-44(s0)
412      fmul.s   fa5,fa4,fa5
413      flw      fa4,-24(s0)
414      fadd.s   fa5,fa4,fa5
415      fsw      fa5,-24(s0)
416      lw      a5,-32(s0)
417      slli     a5,a5,2
418      lw      a4,-52(s0)
419      add      a5,a4,a5
420      flw      fa4,0(a5)
421      flw      fa5,-48(s0)
422      fmul.s   fa5,fa4,fa5
423      flw      fa4,-28(s0)
424      fadd.s   fa5,fa4,fa5
425      fsw      fa5,-28(s0)
426      lw      a5,-32(s0)
427      addi     a5,a5,1
428      sw      a5,-32(s0)
429  .L53:
430      lw      a4,-32(s0)
431      lw      a5,-64(s0)
432      blt      a4,a5,.L54
433      ## lw      a5,-20(s0)
434      ## slli     a5,a5,2

```



```

435      ## lw      a4,-56(s0)
436      ##add      a5,a4,a5
437      ## flw     fa5,-24(s0)
438      ##fsw      fa5,0(a5)
439      #lw        a5,-20(s0)
440      #slli      a5,a5,2
441      #lw        a4,-60(s0)
442      #add       a5,a4,a5
443      #flw      fa5,-28(s0)
444      #fsw      fa5,0(a5)
445      lw        t0,-20(s0)
446      slli      t1,t0,2
447      lw        a4,-56(s0)
448      add       a5,a4,t1
449      flw      fa5,-24(s0)
450      fsw      fa5,0(a5)
451      lw        a4,-60(s0)
452      add       a5,a4,t1
453      flw      fa5,-28(s0)
454      fsw      fa5,0(a5)
455      lw        a5,-20(s0)
456      addi      a5,a5,1
457      sw        a5,-20(s0)
458  .L52:
459      lw        a4,-20(s0)
460      lw        a5,-64(s0)
461      blt       a4,a5,.L55
462      nop
463      nop
464      lw        ra,60(sp)
465      lw        s0,56(sp)
466      addi      sp,sp,64
467      jr        ra

```

3.4)

Pedia para calcular a DFT dos seguintes vetores $x[n]$, com $N=8$

x1: .float 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

$1.0 + i*0.0$

$1.0 + i*0.0$

$1.0 + i*0.0$

$1.0 + i*0.0$

$1.0 + i*0.0$

$1.0 + i*0.0$

$1.0 + i*0.0$

$1.0 + i*0.0$

x2: .float 1.0, 0.7071, 0.0, -0.7071, -1.0, -0.7071, 0.0, 0.7071

$-5.9604645E-8 + i*0.0$

$3.999981 + i*-8.940697E-8$

$2.9587164E-7 + i*5.9604645E-8$

$1.9580126E-5 + i*2.3841858E-7$

$0.0 + i*8.4292884E-7$

$1.8656254E-5 + i*6.854534E-7$

$-6.166414E-7 + i*-5.9604645E-8$

$3.99998 + i*-3.0696392E-6$

x3: .float 0.0, 0.7071, 1.0, 0.7071, 0.0, -0.7071, -1.0, -0.7071

$-5.9604645E-8 + i*0.0$

$-8.940697E-8 + i*-3.9999807$

$-2.2726046E-7 + i*1.7881393E-7$

$-3.5762787E-7 + i*1.886487E-5$

$0.0 + i*-1.1801004E-6$

$-8.6426735E-7 + i*-1.9788742E-5$

$5.6604324E-7 + i*4.7683716E-7$

$2.7120113E-6 + i*3.9999814$

x4: .float 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0

$4.0 + i*0.0$

$0.99999994 + i*-2.4142137$

$-1.5663646E-7 + i*0.0$

$1.0000001 + i*-0.41421354$

$0.0 + i*-2.3841858E-7$

$1.0000004 + i*0.41421318$

$2.0218863E-7 + i \cdot 2.9802322E-7$

$1.00000008 + i \cdot 2.4142137$

3.5)

Pedia para determinados sinais:

3.5.1)

Medir o tempo de execução do procedimento DFT e calcule a frequência do processador RISC-V Uniciclo simulado pelo Rars.

$\text{Texec (s)} = \text{Número de instruções (instret 3074)} \times \text{CPI (é 1)} \times \text{Duração do ciclo (é o inverso da frequência)}$

```
1 # --- Medi o de Desempenho (In cio) ---
2 # 0 PDF usa o endere o 3073 (0xC01) para o registrador 'time'
3 csrr s1, 3074
4 csrr s5, 3073 # s5 = tempo atual (in cio)
5
6 # --- Prepara e chama a DFT ---
7 la a0, x # a0 <- Endere o do vetor de entrada x
8 la a1, X_real # a1 <- Endere o do vetor de resultado
9 la a2, X_imag # a2 <- Endere o do vetor de resultado
10 lw a3, N # a3 <- N mero de pontos N
11 jal DFT
12
13 # --- Medi o de Desempenho (Fim) ---
14 csrr t0, 3073
15 csrr t1, 3074 # t0 = tempo atual (fim)
16 sub s5, t0, s5 # s5 = tempo_fim - tempo_inicio (Tempo
17 de execu o em ms)
18
19 # Imprimir o tempo de execu o
20 mv a0, s5
21 li a7, 1 # Syscall para imprimir inteiro
22 ecall
23 la a0, newline
24 li a7, 4
25 ecall
```

Nesse caso utilizamos o s5 para guardar o valor do tempo de execução em ms, e o s1 é o número de instruções I executadas. Como o item pedia o tempo de execução apenas do procedimento DFT, foi utilizado um breakpoint no mv a0, s5, para ver o número de instruções executadas antes dele. E com esses dois dados obtidos, pegamos o s1 e dividimos pelo s5 para obter a frequência.

a) N=8, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0

s1 = 30661

s5 = 1530

frequência = 20,039869281045751633986928104575

b) N=12, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 71947

s5 = 3600

frequência = 19,985277777777777777777777777778

c) N=16, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 135399

s5 = 6955

frequência = 19,467864845434938892882818116463

d) N=20, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 213265

s5 = 10845

frequência = 19,664822498847395112955278930383

e) N=24, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 308649

s5 = 18975

frequência = 16,266086956521739130434782608696

f) N=28, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 425519

s5 = 29720

frequência = 14,317597577388963660834454912517

g) N=32, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 658589

s5 = 36164

frequência = 18,21117686096670722265236146444

h) N=36, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 723109

s5 = 50415

frequência = 14,343132004363780620846970147773

i) N=40, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0

s1 = 887941

s5 = 75296

frequência = 11,792671589460263493412664683383

j) N=44, x[n]=1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0 , ..., 0.0

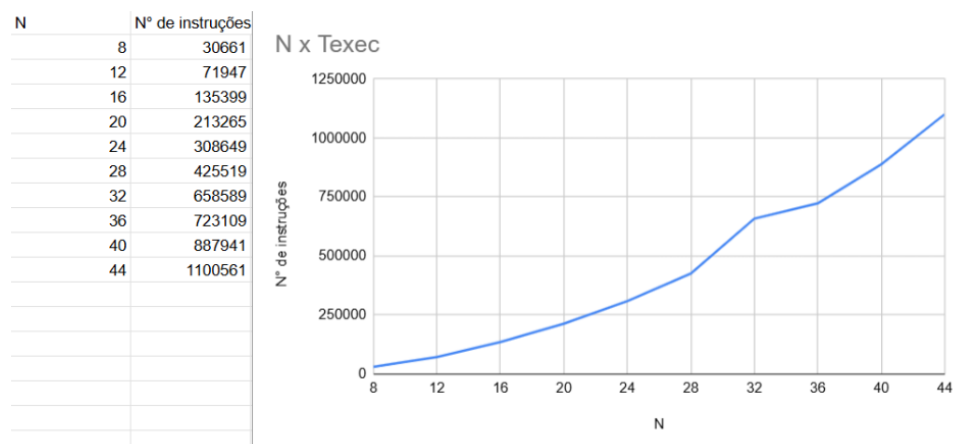
s1 = 1100561

s5 = 87407

frequência = 12,591222670953127323898543594907

3.5.2)

Pedia para fazer um gráfico em escala de N x texec e que conclusões podemos tirar desta análise do gráfico.



É possível observar nesse gráfico um crescimento não linear, ou seja, temos uma curva, ela se mostra cada vez mais íngreme. Isso indica que o tempo de execução aumenta a uma taxa muito maior do que o aumento de N. Além disso tem uma complexidade de $O(N^2)$, pois o tempo de execução cresce de forma quadrática de acordo com a entrada.

Link do vídeo: <https://youtu.be/GuSHfYYWPgc>