Microprocessadores e Microcontroladores (27146)

Hardware *vs* Software

Prof. Ricardo Menotti (menotti@ufscar.br)

Atualizado em: 26 de abril de 2021

Departamento de Computação
Centro de Ciências Exatas e de Tecnologia
Universidade Federal de São Carlos

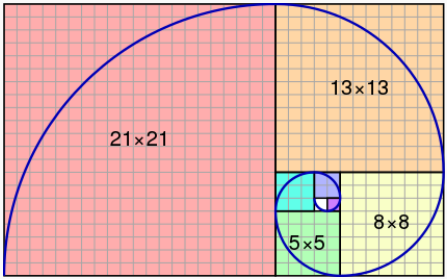
Conteúdo

A sequência de Fibonacci

A sequência de Fibonacci [MatsIsFun.com()]

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

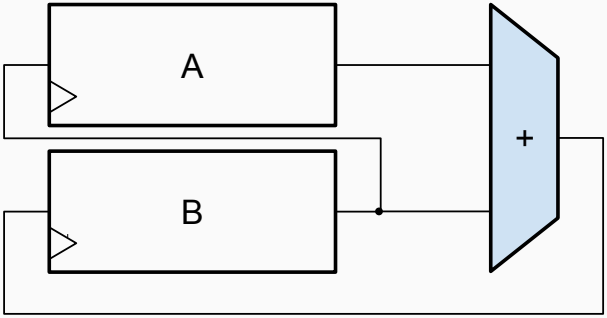


Leonardo Pisano Bogollo

- Fibonacci era seu apelido, que significa “Filho de Bonacci”.
- Seu nome verdadeiro era **Leonardo Pisano Bogollo**, e ele viveu entre 1170 e 1250 na **Itália**.
- Além de ser famoso pela sequência de Fibonacci, ele ajudou a espalhar os numerais hindu-arábicos (como nossos números atuais 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) pela Europa no lugar dos numerais romanos (I, II, III, IV, V, etc).
- O Dia de Fibonacci é **23 de novembro**, pois possui os dígitos “1, 1, 2, 3” que fazem parte da sequência.

Hardware é essencialmente espacial

Uma solução possível

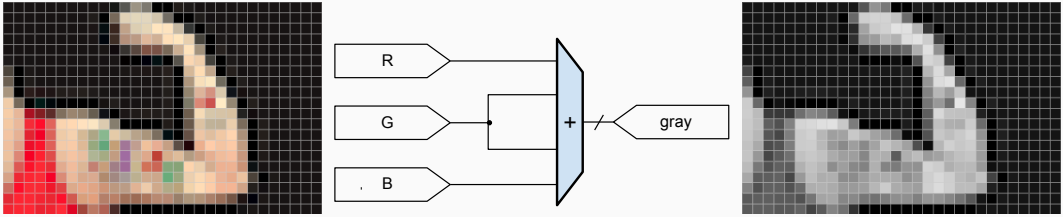



```
1  module fibonacci(
2    input clk,
3    output [31:0] fibo);
4
5    integer a = 0, b = 1;
6
7    always@(posedge clk)
8     begin
9        a <= b;
10       b <= a + b;
11     end
12
13   assign fibo = a;
14 endmodule
```

Um problema paralelizável

$$gray = (R \times 0.299) + (G \times 0.587) + (B \times 0.114)$$

```
1 module gray(
2   input  [23:0] i_rgb,
3   output [23:0] o_rgb);
4   wire [9:0] rgb;
5   assign rgb = (i_rgb[23:16] + (i_rgb[15:8]<<1) + i_rgb[7:0])>>2;
6   assign o_rgb = {rgb[7:0], rgb[7:0], rgb[7:0]};
7 endmodule
```



https://github.com/menotti/convert_grayscale

Software é essencialmente temporal

Processador: hardware de propósito geral

- Máquina com instruções genéricas (lógicas, aritméticas, etc.);

Processador: hardware de propósito geral

- Máquina com instruções genéricas (lógicas, aritméticas, etc.);
- Programa escrito em **linguagem de alto nível**;

Processador: hardware de propósito geral

- Máquina com instruções genéricas (lógicas, aritméticas, etc.);
- Programa escrito em **linguagem de alto nível**;
- Compilador traduz para a **linguagem da máquina**;

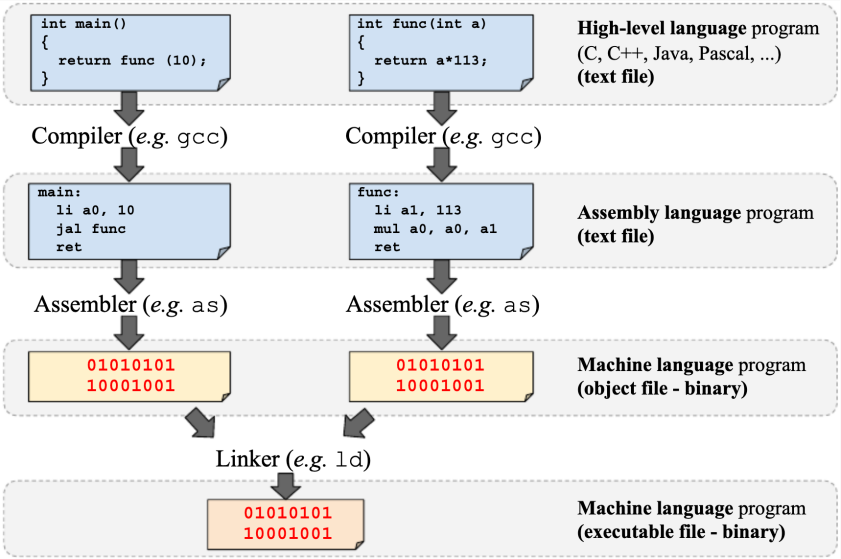
Processador: hardware de propósito geral

- Máquina com instruções genéricas (lógicas, aritméticas, etc.);
- Programa escrito em **linguagem de alto nível**;
- Compilador traduz para a **linguagem da máquina**;
- Sistema operacional **carrega** programa na memória;

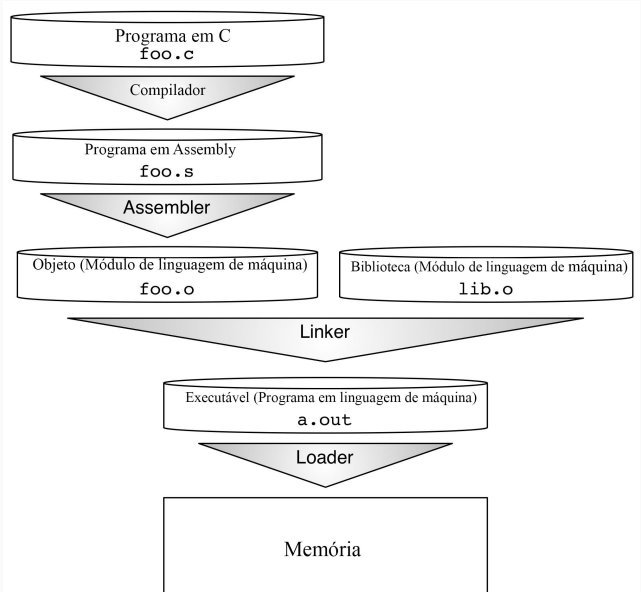
Processador: hardware de propósito geral

- Máquina com instruções genéricas (lógicas, aritméticas, etc.);
- Programa escrito em **linguagem de alto nível**;
- Compilador traduz para a **linguagem da máquina**;
- Sistema operacional **carrega** programa na memória;
- Processador busca **uma instrução de cada vez** e executa.

Geração de um executável nativo [Borin(2021)]



O ciclo de vida de um programa



Sequência de Fibonacci

```python
def fibo(n):
    x, y = 0, 1;
    for i in range(n):
        x, y = y, x + y;
    return x
```

```java
int fibo(int n) {
    int x = 0;
    int y = 1;
    int z;
    for (int i=0; i<n; i++) {
        z = x;
        x = x + y;
        y = z;
    }
    return y;
}
```

ou...

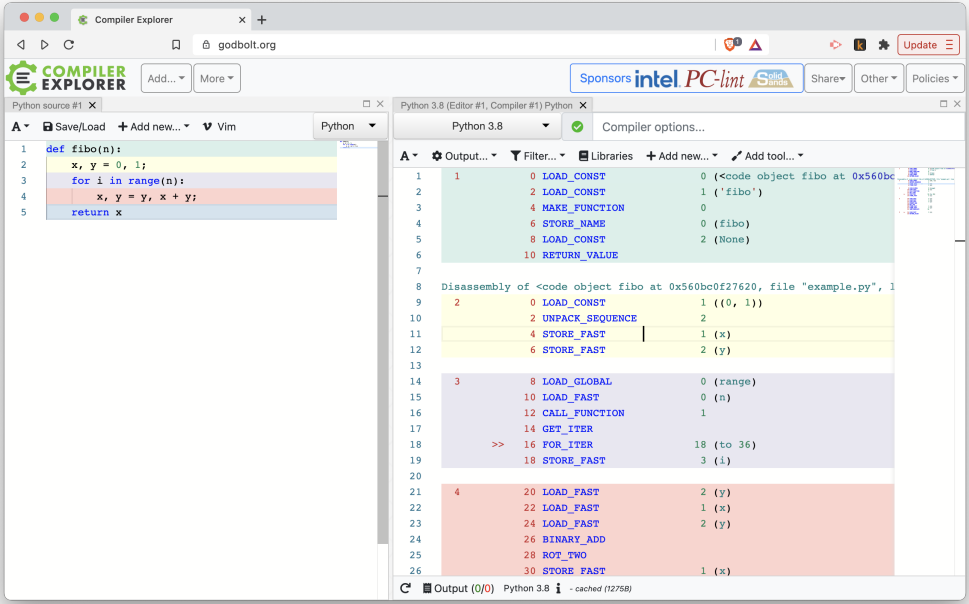
Sequência de Fibonacci

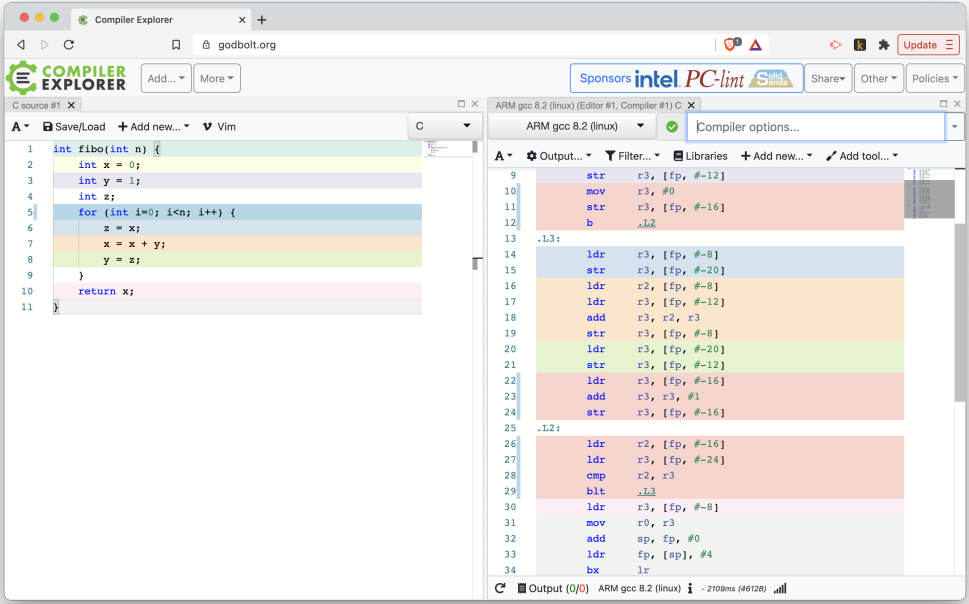
```python
def fibo(n):
    x, y = 0, 1;
    for i in range(n):
        x, y = y, x + y;
    return x
```

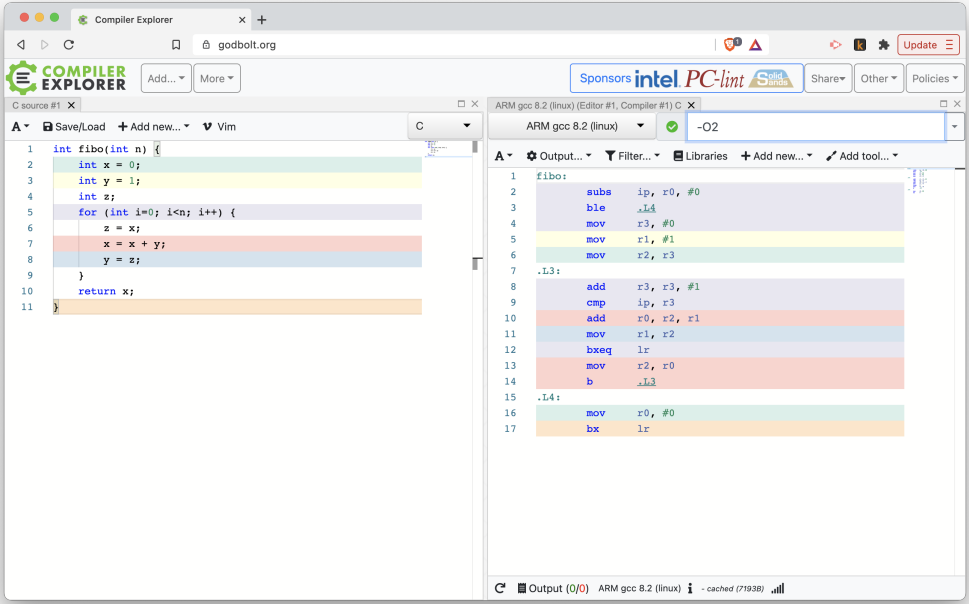
```c
int fibo(int n) {
    int x = 0;
    int y = 1;
    int z;
    for (int i=0; i<n; i++) {
        z = x;
        x = x + y;
        y = z;
    }
    return y;
}
```

ou...

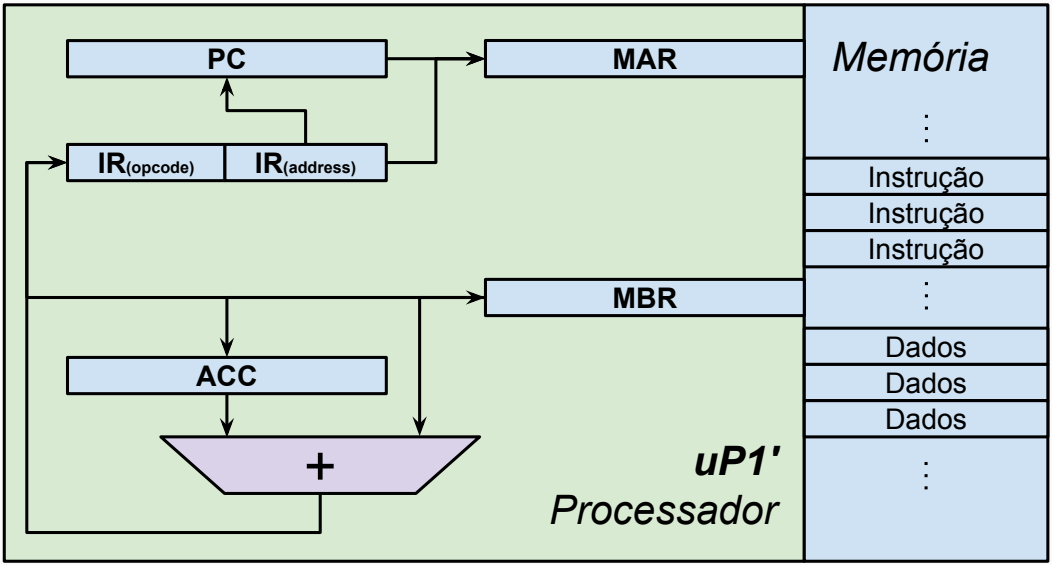
```c
int fibo_rec(int n) {
    if (n < 3)
        return n-1;
    else
        return fibo_rec(n-1) + fibo_rec(n-2);
}
```


Compiler Explorer
godbolt.org
Update
COMPILER EXPLORER
Add...
More
Sponsors intel PC-lint
Share
Other
Policies
Python source #1
Save/Load
Add new...
Vim
Python
def fibo(n):
x, y = 0, 1;
for i in range(n):
x, y = y, x + y;
return x
Python 3.8 (Editor #1, Compiler #1) Python
Python 3.8
Compiler options...
Output...
Filter...
Libraries
Add new...
Add tool...
1 0 LOAD_CONST 0 (<code object fibo at 0x560bc
2 LOAD_CONST 1 ('fibo')
4 MAKE_FUNCTION 0
6 STORE_NAME 0 (fibo)
8 LOAD_CONST 2 (None)
10 RETURN_VALUE
Disassembly of <code object fibo at 0x560bc0f27620, file "example.py", l
2 0 LOAD_CONST 1 ((0, 1))
2 UNPACK_SEQUENCE 2
4 STORE_FAST 1 (x)
6 STORE_FAST 2 (y)
3 8 LOAD_GLOBAL 0 (range)
10 LOAD_FAST 0 (n)
12 CALL_FUNCTION 1
14 GET_ITER
>> 16 FOR_ITER 18 (to 36)
18 STORE_FAST 3 (i)
4 20 LOAD_FAST 2 (y)
22 LOAD_FAST 1 (x)
24 LOAD_FAST 2 (y)
26 BINARY_ADD
28 ROT_TWO
30 STORE_FAST 1 (x)
Output (0/0)
Python 3.8
- cached (1275B)


Compiler Explorer
godbolt.org
Update
COMPILER EXPLORER
Add...
More
Sponsors intel PC-lint
Share
Other
Policies
C source #1
Save/Load
Add new...
Vim
C
int fibo(int n) {
int x = 0;
int y = 1;
int z;
for (int i=0; i<n; i++) {
z = x;
x = x + y;
y = z;
}
return x;
}
ARM gcc 8.2 (linux) (Editor #1, Compiler #1) C
ARM gcc 8.2 (linux)
Compiler options...
Output...
Filter...
Libraries
Add new...
Add tool...
str r3, [fp, #-12]
mov r3, #0
str r3, [fp, #-16]
b .L2
.L3:
ldr r3, [fp, #-8]
str r3, [fp, #-20]
ldr r2, [fp, #-8]
ldr r3, [fp, #-12]
add r3, r2, r3
str r3, [fp, #-8]
ldr r3, [fp, #-20]
str r3, [fp, #-12]
ldr r3, [fp, #-16]
add r3, r3, #1
str r3, [fp, #-16]
.L2:
ldr r2, [fp, #-16]
ldr r3, [fp, #-24]
cmp r2, r3
blt .L3
ldr r3, [fp, #-8]
mov r0, r3
add sp, fp, #0
ldr fp, [sp], #4
bx lr
Output (0/0)
ARM gcc 8.2 (linux)
- 2109ms (4612B)


Compiler Explorer
godbolt.org
Update
COMPILER EXPLORER
Add...
More
Sponsors intel PC-lint
Share
Other
Policies
C source #1
Save/Load
Add new...
Vim
C
int fibo(int n) {
int x = 0;
int y = 1;
int z;
for (int i=0; i<n; i++) {
z = x;
x = x + y;
y = z;
}
return x;
}
ARM gcc 8.2 (linux) (Editor #1, Compiler #1) C
ARM gcc 8.2 (linux)
-O2
Output...
Filter...
Libraries
Add new...
Add tool...
fibo:
subs ip, r0, #0
ble .L4
mov r3, #0
mov r1, #1
mov r2, r3
.L3:
add r3, r3, #1
cmp ip, r3
add r0, r2, r1
mov r1, r2
bxeq lr
mov r2, r0
b .L3
.L4:
mov r0, #0
bx lr
Output (0/0)
ARM gcc 8.2 (linux)
- cached (71938)

Processador - uP1' (8 bits) [Hamblen and Furman(2001)]



Conjunto de instruções (ISA) do uP1' - 8 bits - 4 instruções - 2 formatos

- Formato M
  - `Endereço: 0x1111_endereço_D`

7	6	5	4	3	2	1	0	*bits*
0	0	0	0	endereço D				*não usada*
0	0	0	1	endereço D				*não usada*
0	0	1	0	endereço D				*não usada*
0	0	1	1	endereço D				STORE
0	1	0	0	endereço D				LOAD
0	1	0	1	endereço D				ADD
0	1	1	0	endereço D				*não usada*
0	1	1	1	endereço D				*não usada*

- Formato J
  - `Endereço: 0x0_endereço_I`

7	6	5	4	3	2	1	0	*bits*
1	endereço I (+7 bits)							JUMP

Organização de memória do uP1'

- Instruções

endereço								*palavras*
0	0	0	0	0	0	0	0	
				.				
				.				$2^7 = 128$
				.				
0	1	1	1	1	1	1	1	

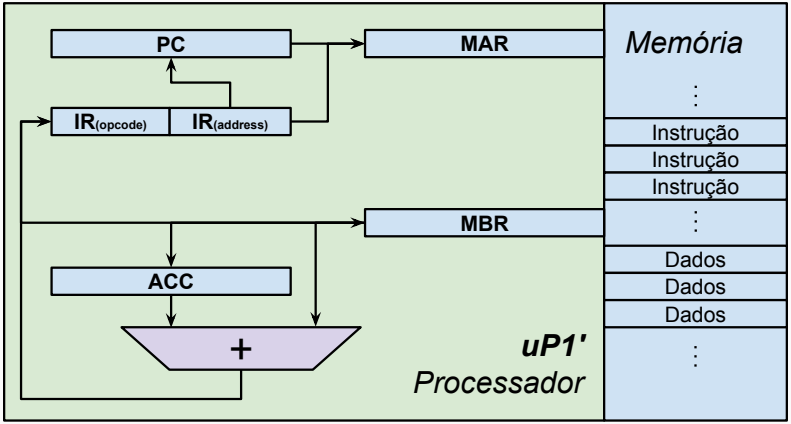
- Não usada

endereço								*palavras*
1	0	0	0	0	0	0	0	
				.				$2^7 - 2^4 = 112$
1	1	1	0	1	1	1	1	

- Dados

endereço								*palavras*
1	1	1	1	0	0	0	0	
				.				$2^4 = 16$
1	1	1	1	1	1	1	1	

Processador - uP1'

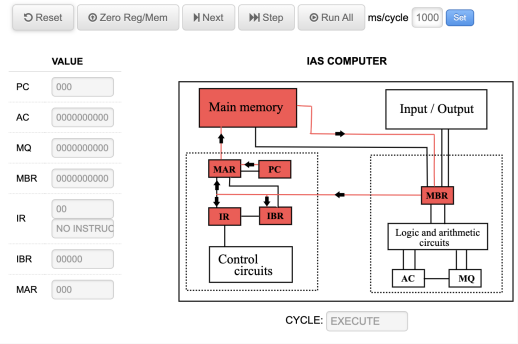


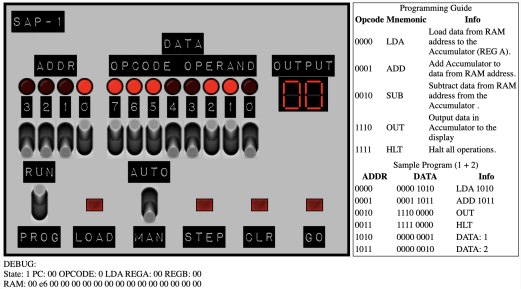

```
0   41   0100 0001   LOAD  A
1   52   0101 0010   ADD   B
2   33   0011 0011   STORE C
3   42   0100 0010   LOAD  B
4   31   0011 0001   STORE A
5   43   0100 0011   LOAD  C
6   32   0011 0010   STORE B
7   80   1000 0000   JUMP  0
    ...
240 FF   1111 1111   // Dados
241 00   0000 0000   // A
242 01   0000 0001   // B
243 00   0000 0000   // C
```

Experimente ele aqui!

Referências

Outros processadores didáticos





- IAS
  - Instruções
  - Simulador
- SAP-1
  - Instruções
  - Simulador

Para saber mais...

- How does a computer do recursion from a hardware standpoint?
- Tail recursion in Hardware
- Recursion in Hardware: Applicability and Implementation Strategies
- Recursive functional hardware descriptions using CλaSH
- Mapping recursive functions to reconfigurable hardware

Bibliografia i

Edson Borin.

An Introduction to Assembly Programming with RISC-V.

2021.

URL `https://riscv-programming.org/`.

J. O. Hamblen and M. D. Furman.

Rapid Prototyping of Digital Systems.

Kluwer, 2001.

Bibliografia ii

MatsIsFun.com.

Fibonacci sequence, 2021.

URL https://www.mathsisfun.com/numbers/fibonacci-sequence.html.

Matt Godbolt.

Compiler explorer, 2021.

URL https://godbolt.org/.