

Exercício Prático 5:

Felipe Rivetti Mizher – 821811

Experiência 1:

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	2,462944s	2,526092	2,526092s	2,652380s	2,652380	2,652380	2,841596s
int	2,715532s	2,841820s	2,778672	3,031260s	3,094408s	3,094408s	3,599576s
float	3,220696	12,437828s	XXXX	10,356400	12,690416	XXXX	10,608752s

Tipo	MIPS (ATM328P)					
	Constante (Ex.: i=i op 3 ;)			Variável (Ex.: i=i op j ;)		
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	15,8358	15,8358	5,2788	5,2788	5,2788	2,6409
int	7,9184	15,8378	3,1672	3,6393	3,6393	1,1311

Tipo	MFLOPS (ATM328P)					
	Constante			Variável		
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
float	0,1084	XXXXXXXX	0,1401	0,1055	XXXXXX	0,1353

Tipo	CPI					
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	4,0417	4,0417	4,2438	4,2438	4,2438	4,5465
Int	4,5469	4,4458	4,8500	4,9510	4,9510	5,7593
float	147,4741	XXXX	118,2088	151,5155	XXXX	118,2088

Experiência 2:

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
char	7.6	18.8	18.8	25.6	18.8	18.9	25.8
Int	7.6	7.6	7.6	10.2	7.6	7.7	9.9
float	7.6	9.5	XXXXXX	12.1	8.9	XXXXXX	10.9

Tipo	MIPS (Seu PC)					
	Constante			Variável		
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
char	892.8	892.8	555.5	892.8	884.9	549.4
int	0	0	3.8	0	100.0	4.3

Tipo	MFLOPS (Seu PC)					
	Constante			Variável		
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
float	5.2	XXXXXX	2.2	7.6	XXXXXX	3.0

Tipo	CPI					
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	8.4	8.4	13.5	8.4	8.47	13.6
Int	0	0	1.95	0	0.075	1.72
float	1.42	XXXXXX	3.37	0.97	XXXXXX	2.47

Comparando diferentes Máquinas:

Identificação da máquina (processador, frequência de clock, SO e Compilador usado)	Prog. em C		Performance Test	
	Speed up (inteiros)	Speed up (FP)	Speed up (inteiros)	Speed up (FP)
MackBook Air 10.1 - Apple M1 (8cores @ 3.2GHz, macOS, clang - 02)	1	1	1	1
Intel Core i5-10300H @ 2.5GHz	1.28x	2.04x	1.74x	1.41x
MackBook Air - Apple M4 (10 cores, 16 GB, macOS, clang - 02)	1.56x	1.56	0.90x	0.49x

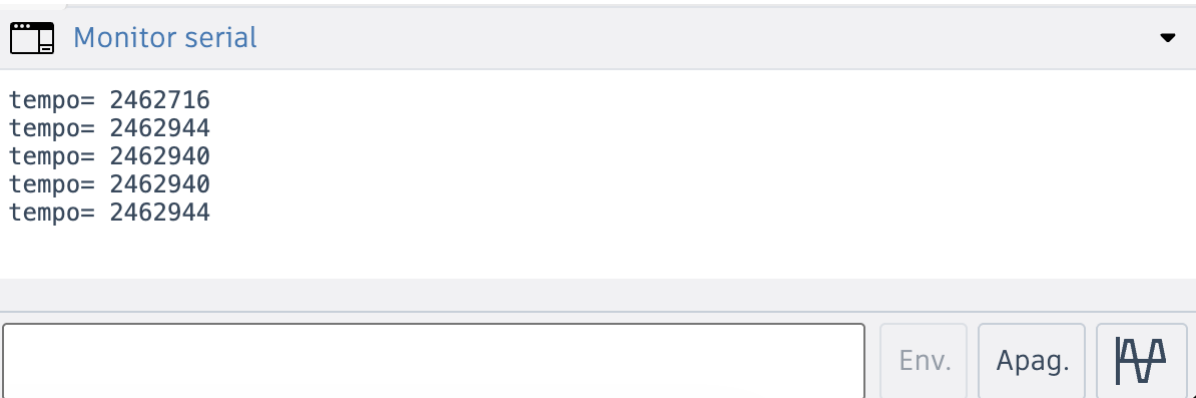
Teste por SO, Compilador e Processador:

Identificação do processador, frequência de clock, compilador	Prog. em C (inteiros)		Speed up
	Escreva aqui o SO utilizado	Escreva aqui o SO utilizado	
Apple M1(8 cores) - 3200MHz - clang 02	MacOS Monterey 14	MacOS Monterey 14	1.00x
Apple M4(10 cores, AMR64) - 7500MHz - clang 02	MacOS Sequoia 15	MacOS Sequoia 15	1.56x
Intel Core i5 - 10300H @ 2.50GHz - 2496.1MHz - GCC	Windows 11 Home	Windows 11 Home	1.28x

Identificação do processador, frequência de clock, SO	Prog. em C (inteiros)		Speed up
	<i>Escreva aqui o Compilador utilizado</i>	<i>Escreva aqui o Compilador utilizado</i>	
Apple M1(8 cores) - 3200MHz - Monterey 14	clang -02	clang -02	1.00x
Apple M4(10 cores, AMR64) - 7500MHz - Sequoia 15	clang -02	clang -02	1.56x
Intel Core i5 - 10300H @ 2.50GHz - 2496.1MHz - GCC	GCC 14.2.0	GCC 14.2.0	1.28x

Identificação do SO e Compilador	Prog. em C (inteiros)		Speed up
	<i>Detalhes da Máquina</i>	<i>Detalhes da Máquina</i>	
MacOS Monterey 14 (M1) - clang - O2	Apple M1 (8 cores, 3200MHz)	MacOS Monterey 14	1.00x
MacOS Sequoia (M4) - clang - O2	Apple M4 (10 cores, 7500MHz)	MacOS Sequoia 15	1.56x
Windows 11 Home - GCC 14.2.0	i5 - 10300H (2496 MHz)	Windows 11 Home	1.28x

Teste no Arduino:



Primeiro Teste Programa em C:

```
Tempo : 18 ms.
Tempo : 10 ms.
Tempo : 7 ms.
Tempo : 7 ms.
Tempo : 6 ms.
Tempo : 6 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo gasto media: 7.4 ms.%
```

Segundo Teste Programa em C:

```
Tempo: 12.026 ms.
Tempo: 10.918 ms.
Tempo: 12.605 ms.
Tempo: 13.997 ms.
Tempo: 13.481 ms.
Tempo: 11.888 ms.
Tempo: 10.858 ms.
Tempo: 10.094 ms.
Tempo: 9.825 ms.
Tempo: 9.643 ms.
Tempo gasto medio: 11.5335 ms.%
```

Primeiro Teste Programa de Benchmark:

Run	Integer Math	56021 Million Operations/s
Run	Floating Point Math	52241 Million Operations/s

Segundo Teste Programa de Benchmark:

Run	Integer Math	31874 Million Operations/s
Run	Floating Point Math	36914 Million Operations/s