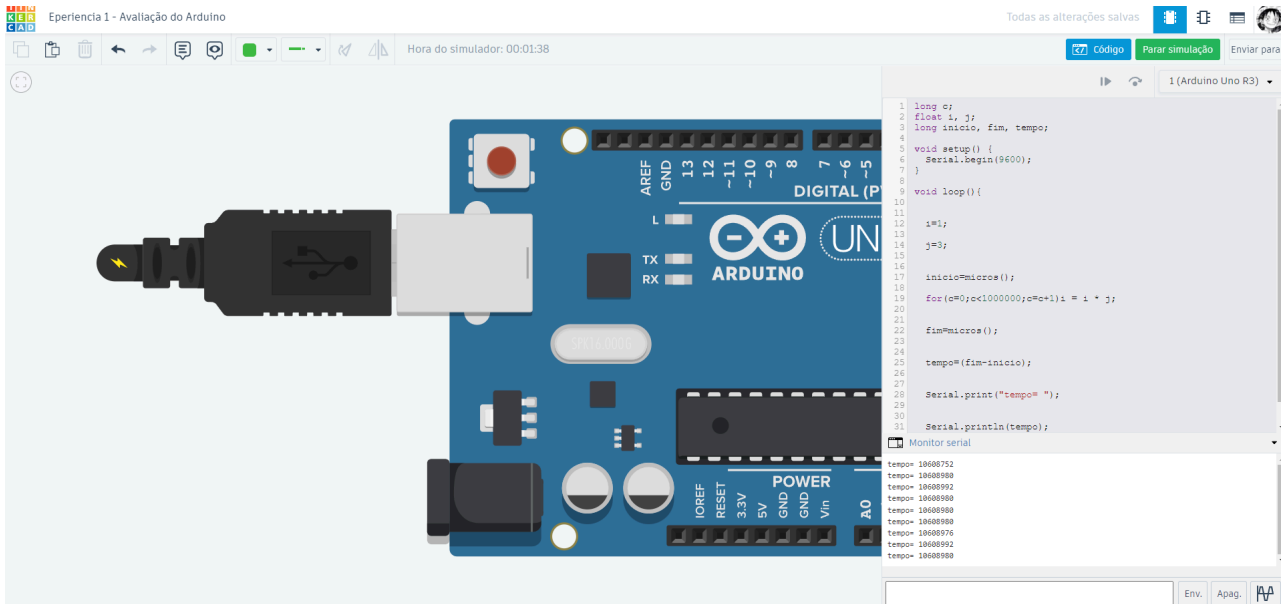


Experiência 1 - Avaliação do Arduino

Teste no Tinkercad Arduino:



- Multiplicação entre floats

1) Tabela baseline (tabela do arduino):

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2462944	2525856	2399568	2652156	2652156	2841596	2841596
int	2715528	2841592	2589232	3031032	3094168	3220468	3599348
float	3220696	12437600	XXXX	10356400	12690176	XXXX	10608752

2) Tabela comparativa MIPS:

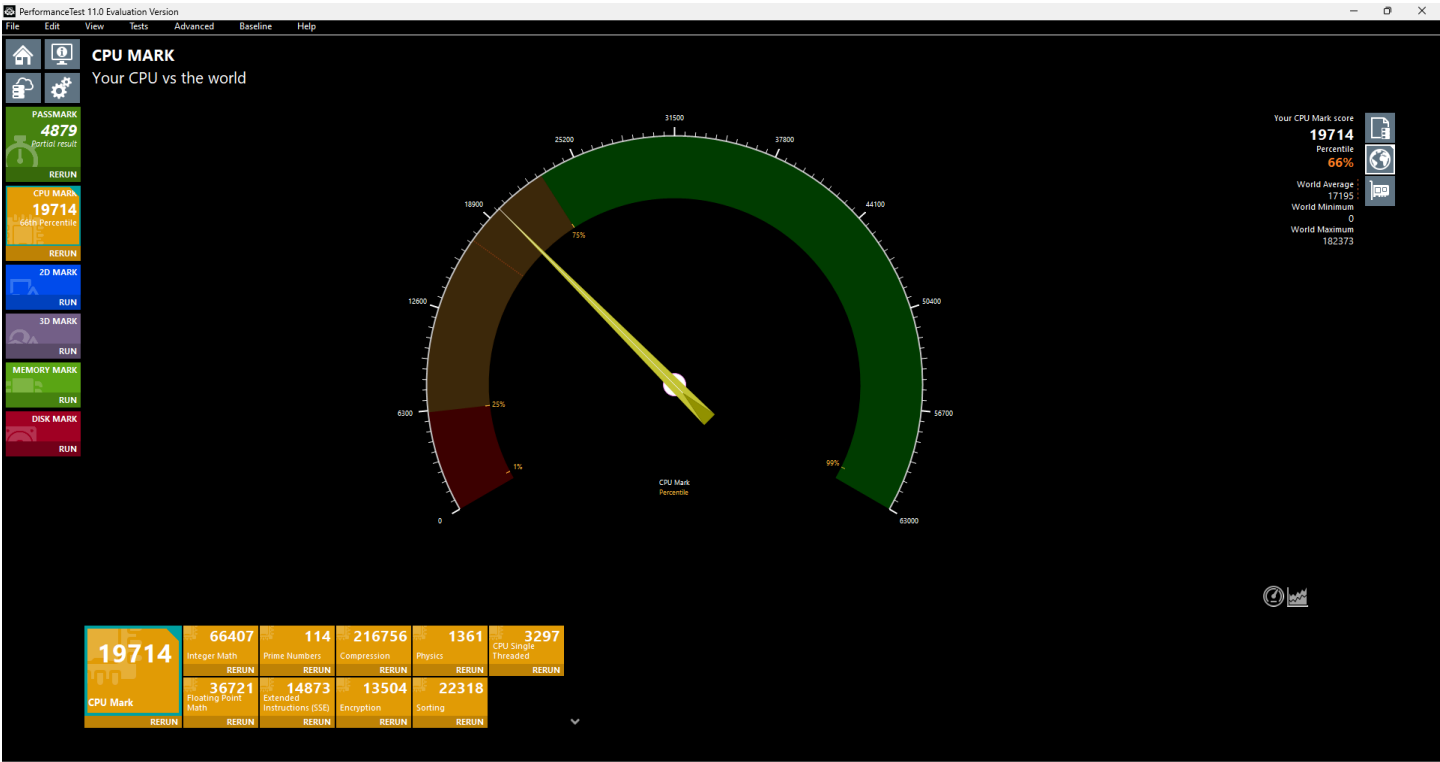
Tipo	MIPS (ATM328P)					
	Constante (Ex.: i=i op 3 ;)			Variável (Ex.: i=i op j ;)		
	Soma	Or	Mult	Soma	Or	Mult
byte	15,8952	15,7788	5,2850	5,2850	2,6409	2,6409
int	7,9324	7,9179	7,9179	2,6410	1,9804	1,1314
Tipo	MFLOPS (ATM328P)					
	Constante			Variável		
	Soma	Or	Mult	Soma	Or	Mult
float	0,1084	XXXXXXX	0,1401	0,1056	XXXXXX	0.1353

3) Tabela CPI:

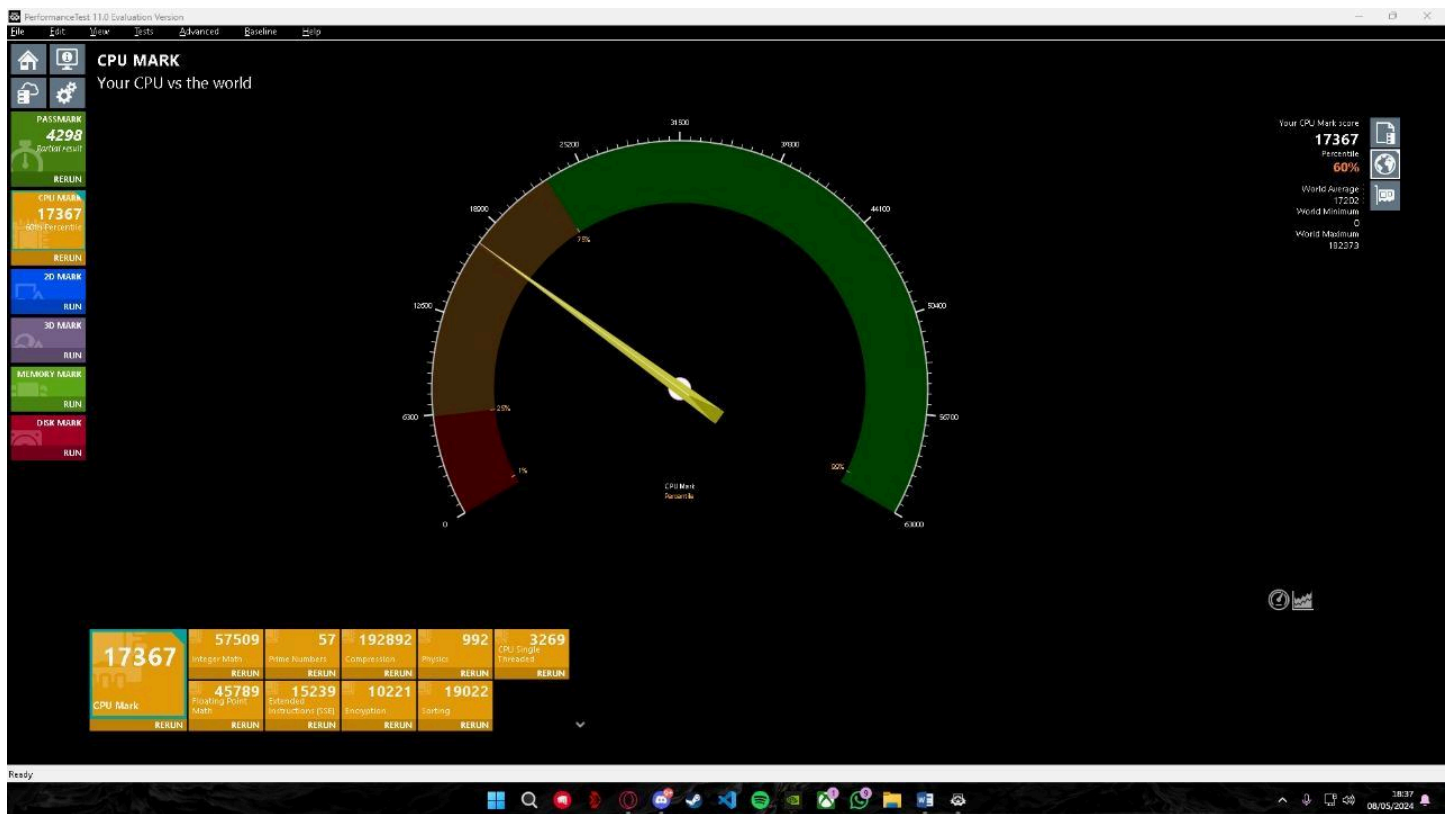
Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
byte	1,0065	1,0140	3,0273	3,0273	6,0584	6,0584
Int	2,0170	2,0170	5,0480	6,0582	8,0790	14,1411
float	147,4704	XXXX	114,1712	151,5116	XXXX	118,2088

Experiência 2 - Seu PC

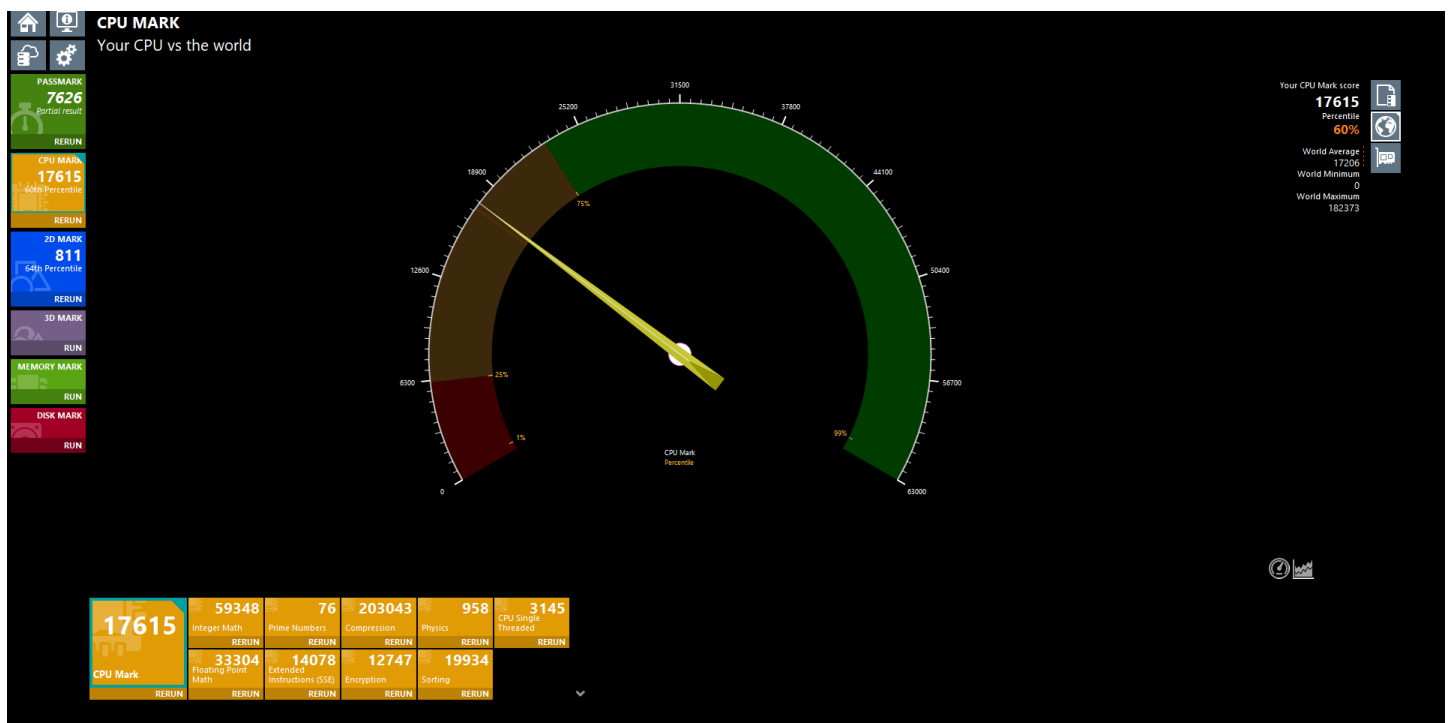
1) Testes de Benchmark com o App:



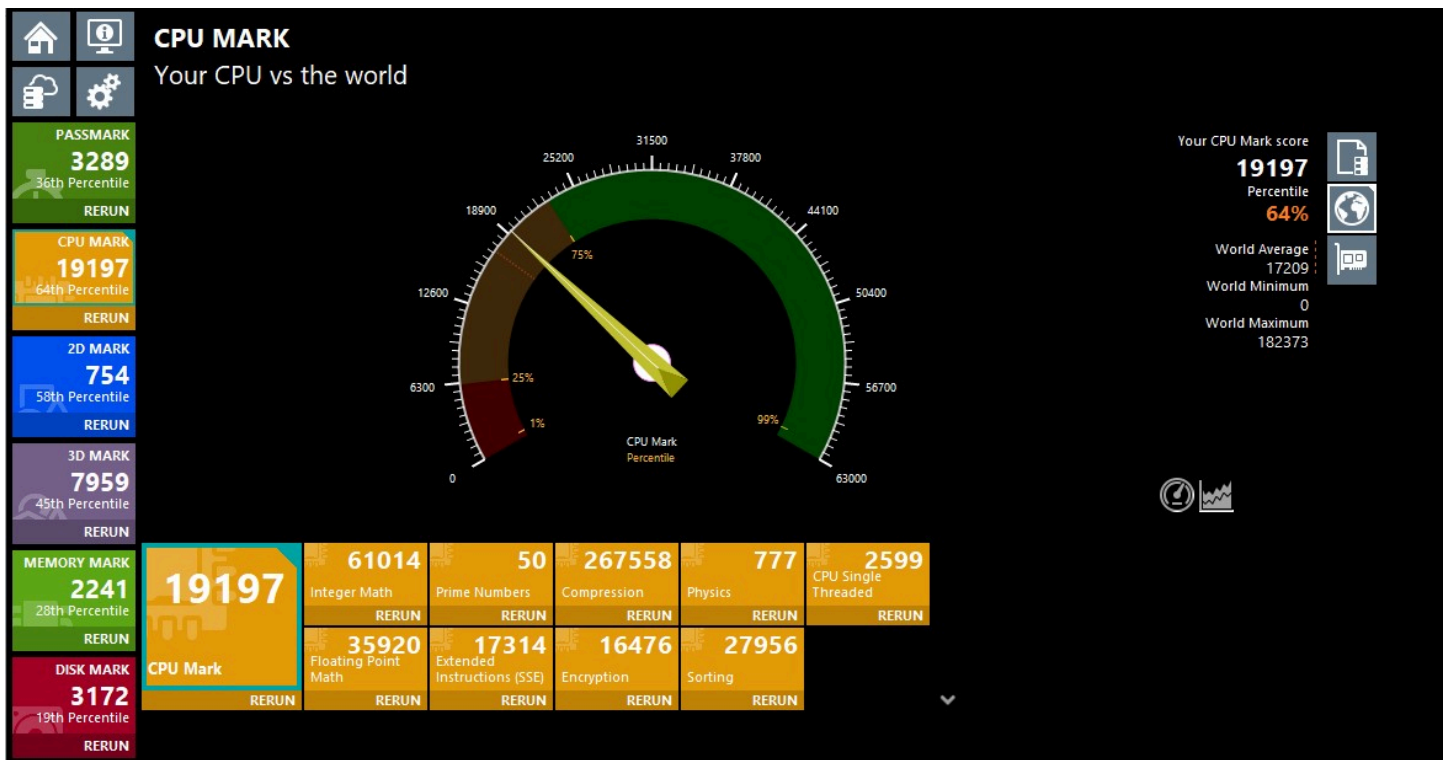
- Performance Test- Gabriel Henrique



- Performance Test- Caio Gomes



- Performance Test- Daniel Salgado



- Performance Test- Arthur Martinho

2) Testes do programa em C:

```
C main.c 3 X
C main.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main()
6 {
7     clock_t inicio, fim, T;
8     float Tempo, media = 0;
9     int c;
10
11     int k;
12
13     char i = 3;
14     char j = 1;
15
16     T = CLOCKS_PER_SEC;
17     for (k = 1; k <= 10; k = k + 1)
18     {
19         inicio = clock();
20         for (c = 1; c <= 10000000; c = c + 1) i = i * j;
21         fim = clock();
22         Tempo = ((fim - inicio) * 1000 / CLOCKS_PER_SEC);
23         printf("\nTempo : %g ms.", Tempo);
24         media = media + Tempo;
25     }
26     printf("\nTempo gasto media: %g ms.", media / 10);
27 }
28
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Executing task: C:\Windows\System32\cmd.exe /d /c .\build\Debug\outDebug.exe

Tempo : 21 ms.
Tempo : 21 ms.
Tempo : 21 ms.
Tempo : 21 ms.
Tempo : 21 ms.
Tempo : 22 ms.
Tempo : 21 ms.
Tempo : 22 ms.
Tempo : 20 ms.
Tempo : 22 ms.
Tempo gasto media: 21.2 ms. Terminal will be reused by tasks, press any key to close it.

- Teste - Gabriel Henrique

```
C performance.c (Untracked) U C performance.c U C main.c 1 U X
teste_C_cpp_arduino > teste_C_cpp_arduino > teste_c > C main.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, I;
8      float Tempo, media = 0;
9      int c;
10
11      int k;
12
13      int i = 3;
14      int j = 1;
15
16      T = CLOCKS_PER_SEC;
17      for (k = 1; k <= 10; k = k + 1)
18      {
19          inicio = clock();
20          for (c = 1; c <= 10000000; c = c + 1) i = i || j;
21          fim = clock();
22          Tempo = ((fim - inicio) * 1000 / CLOCKS_PER_SEC);
23          printf("\nTempo : %g ms.", Tempo);
24          media = media + Tempo;
25      }
26      printf("\nTempo gasto media: %g ms.", media / 10);
27  }
28

PROBLEMS 1 OUTPUT TERMINAL PORTS

Tempo : 4 ms.
Tempo : 5 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo gasto media: 4.4 ms.
PS E:\faculdade\CC_PUC\ARQ2\ARQ2\EPs\EP5\teste_C_cpp_arduino\teste_C_cpp_arduino\teste_c\output>
```

- Teste - Caio Gomes

```
C teste.c 2 X
C teste.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, I;
8      float Tempo, media = 0;
9      register int c;
10
11      int k;
12
13      int i = 3;
14      int j = 1;
15
16      T = CLOCKS_PER_SEC;
17      for (k = 1; k <= 10; k = k + 1)
18      {
19          inicio = clock();
20          for (c = 1; c <= 10000000; c = c + 1) i = j;
21          fim = clock();
22          Tempo = ((fim - inicio) * 1000 / CLOCKS_PER_SEC);
23          printf("\nTempo : %g ms.", Tempo);
24          media = media + Tempo;
25      }
26      printf("\nTempo gasto media: %g ms.", media / 10);
27  }
28

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Tempo : 4 ms.
Tempo : 1 ms.
Tempo : 4 ms.
Tempo : 2 ms.
Tempo : 3 ms.
Tempo : 2 ms.
Tempo : 4 ms.
Tempo : 3 ms.
Tempo : 2 ms.
Tempo gasto media: 2.6 ms.
PS D:\Daniel\Faculdade\Semestre 3\ARQ2\Códigos\output>
```

- Teste - Daniel Salgado

main.c X

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media = 0;
9      register int c;
10
11      int k;
12
13      float i = 1;
14      float j = 3;
15
16      T = CLOCKS_PER_SEC;
17      for (k = 1; k <= 10; k = k + 1)
18      {
19          inicio = clock();
20          for (c = 1; c <= 10000000; c = c + 1) i = i * j;
21          fim = clock();
22          Tempo = ((fim - inicio) * 1000 / CLOCKS_PER_SEC);
23          printf("\nTempo : %g ms.", Tempo);
24          media = media + Tempo;
25      }
26      printf("\nTempo gasto media: %g ms.", media / 10);
27  }
```

C:\Users\User\Desktop\EP5\bin\Debug\EP5.exe

```

Tempo : 26 ms.
Tempo : 27 ms.
Tempo : 26 ms.
Tempo : 27 ms.
Tempo : 26 ms.
Tempo : 26 ms.
Tempo : 26 ms.
Tempo : 27 ms.
Tempo : 26 ms.
Tempo : 27 ms.
Tempo gasto media: 26.4 ms.
Process returned 0 (0x0)   execution time : 0.272 s
Press any key to continue.
```

-Teste -Arthur Martinho

3) Tabelas de cada um preenchidas com as medidas realizadas:

3.1) PC (Gabriel Henrique):

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	5 ms	16.6 ms	3.9 ms	17.7 ms	17.1 ms	7.1 ms	20.4 ms
Int	4.3 ms	5.3 ms	4.5 ms	5 ms	5.2 ms	9.4 ms	6.9 ms
float	4.8 ms	43.7 ms	XXXXXX	43.1 ms	43.1 ms	XXXXXX	43 ms

Tipo	MIPS (Seu PC)					
	Constante			Variável		
	Soma	Or	Mult	Soma	Or	Mult
char	862.06	9090.90	787.40	826.44	4761.90	649.35
int	10000,00	50000,00	14285.71	11111,11	1960.78	3846.15
Tipo	MFLOPS (Seu PC)					
	Constante			Variável		
	Soma	Or	Mult	Soma	Or	Mult
float	257.06	XXXXXX	261.09	261.09	XXXXXX	261.78

Clock Frequencies

Measured Speed

3737.2 MHz [Turbo: 4542.2 MHz]

Tipo	CPI					
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	4.3176	0.4094	4.7270	4.5037	0.7816	5.7320
Int	0.3722	0.0744	0.2605	0.3349	1.8982	0.9677
float	14.4789	XXXXXX	14.2556	14.2556	XXXXXX	14.2184

3.2) PC (Caio Gomes):

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	3.5 ms	25 ms	3.6 ms	5.2 ms	4.2 ms	5.2 ms	7.8 ms
Int	4.1 ms	3.7 ms	4.4 ms	5.2 ms	19 ms	4.7 ms	7.5 ms
float	5.9 ms	25.5 ms	XXXXXX	1060.9 ms	25.7 ms	XXXXXX	28 ms

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	473,93	100.000	5.882,35	14.285,71	5.882,35	2.325,58
int	25.000	33.333,33	9090,90	671,14	16.666,66	2.941,17
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	510,20	XXXXXX	9,42	505,05	XXXXXX	452,48

Clock Frequencies	
Measured Speed	2496.9 MHz [Turbo: 3995.0 MHz]

Tipo	CPI					
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	6,315863	0,025	0,435	0,175	0,425	1,075
Int	0,100	0,075	0,275	3,725	0,150	0,850
float	4,9	XXXXXX	265,225	4,950	XXXXXX	5,525

3.3) PC (Daniel Salgado):

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	2.6ms	17ms	2.4ms	18.6ms	17.2ms	4.5ms	19.8ms
Int	2.6ms	2.2ms	2.4ms	4.6ms	2.4ms	7ms	7ms
float	3.2ms	44.4ms	XXXXXX	44.4ms	44.1ms	XXXXXX	44.5ms

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	694,444	-50000	625	684,93	5263,15	581,3953
int	-25000	-50000	5000	-50000	2272,7272	2272,7272
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	242,71844	XXXXXX	242,71844	244,49877	XXXXXX	242,13075

Clock Frequencies

Measured Speed

Multiplier

3675.4 MHz [Turbo: 4517.4 MHz]

36.6X

Tipo	CPI					
	<i>Soma</i>	<i>Or</i>	<i>Mult</i>	<i>Soma</i>	<i>Or</i>	<i>Mult</i>
byte	5,292576	-0.073508	5,88064	5,366084	0.698326	6,321688
Int	0,147016	0,073508	0,73508	0,073508	1,617176	1,617176
float	15,142648	XXXXXX	15,142648	15,032386	XXXXXX	15,179402

3.4) PC (Arthur Martinho):

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	2,4 ms	19,2 ms	2,4 ms	21,5 ms	19,2 ms	7,4 ms	23,8 ms
Int	2,4 ms	16,8 ms	10 ms	21,6 ms	16,8 ms	7,4 ms	24 ms
float	2,5 ms	26,4 ms	xxxxxx	26,5 ms	26,4 ms	xxxxxx	26,6 ms

Tipo	MIPS (Seu PC)					
	Constante			Variável		
	Soma	Or	Mult	Soma	Or	Mult
char	595,238	100000	523,560	595,238	2000	467,289
int	694,444	1315,789	520,833	694,444	2000	462,962

Tipo	MFLOPS (Seu PC)					
	Constante			Variável		
	Soma	Or	Mult	Soma	Or	Mult
float	418,410	xxxxxx	416,666	418,410	xxxxxx	414,937

Clock Frequencies		
Measured Speed	3600.6 MHz	3696.2 MHz [Turbo: 4395.5 MHz]
Multiplier	(N/A)	37.0X

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
byte	5,041	0,030	5,731	5,041	1,5003	6,421
Int	4,4408	2,280	5,761	4,4408	1,5003	6,481
float	7,171	xxxxxx	7,201	7,171	xxxxxx	7,231

4) Tabelas de comparação com o Speedup:

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)
12th Gen Intel(R) Core(TM) i5-12400F - 2.50 GHz - 2496.9 - Windows - gcc	1	1	1	1
AMD Ryzen 7 3700X 8-Core Processor - 3.60 GHz - 3600.6 MHz - Windows - gcc	2.16	1.0037	3.2432	1.0075
AMD Ryzen 5 5600X 6-Core Processor - 3.70 GHz - 3737.7 MHz - Windows - gcc	1.1777	1.0139	3.5252	2.8178
AMD Ryzen 5 5600X 6-Core Processor - 3.70 GHz - 3675.4 MHz - Windows - gcc	2.0909	1.2717	2.9166	3.9952

Identificação do processador, frequência de clock, compilador	Prog. em C (inteiros)		Speed up
	Windows 11	Windows 10	
12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz - 2496.9 MHz - gcc	848.52	818.13	1.0371

Identificação do processador, frequência de clock, SO	Prog. em C (inteiros)		Speed up
	gcc	MSVC	
AMD Ryzen 7 3700X 8-Core Processor - 3.60 GHz - 3600.6 MHz - Windows	2970.91	848.52	3.5013

Identificação do SO e Compilador	Prog. em C (inteiros)		Speed up
	AMD Ryzen 5 5600X 6- Core Processor 3.70 GHz - 3737.7 Mhz	AMD Ryzen 5 5600X 6- Core Processor 3.70 GHz - 3675.4 Mhz	
Windows - gcc	848.52	17137.20	20.1965

