



Das metodologias ativas ao ensino remoto: ferramentas e experiências

Ensino prático a distância na computação

Prof. Ricardo Menotti <menotti@ufscar.br>

13 de agosto de 2020

SEMANA DO ENSINO REMOTO PARA AS CIÊNCIAS EXATAS E TECNOLÓGICAS

Departamento de Computação

Centro de Ciências Exatas e de Tecnologia

Universidade Federal de São Carlos

Conteúdo

1. Presencial

Contexto

Metodologia

Práticas

2. Remoto

Contexto

Metodologia

Ferramentas

3. Galeria

Dicas

Presencial

Disciplina: **Lógica Digital**

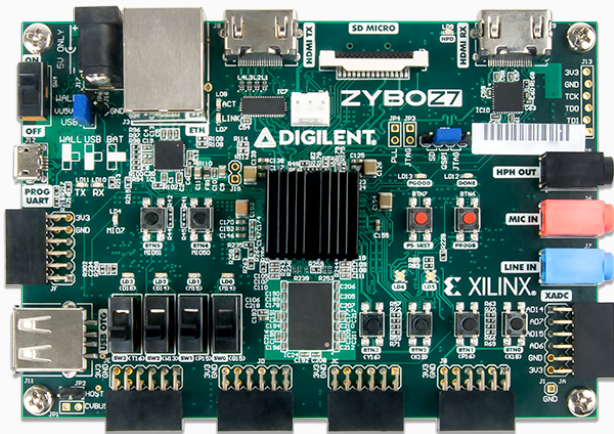
Diferentes abordagens/tecnologias/linguagens
Dificuldade de continuidade nas disciplinas seguintes

- **2º/3º semestres**
- 90+ estudantes
 - 45+ por sala teórica
 - 26 por laboratório
- **1º semestre**
- 170+ estudantes
 - 85+ por sala teórica
 - 26+ por laboratório

- 3º semestre
 - 90+ estudantes
 - 45+ por sala teórica
 - 26 por laboratório
 - aulas expositivas
 - minitestes
 - exercícios (opcionais)
 - provas
 - laboratórios
- 1º semestre
 - 170+ estudantes
 - 85+ por sala teórica
 - 26 por laboratório
 - aulas gravadas
 - questionários (online)
 - exercícios (grupo)
 - provas (*orais!?)
 - simulações*

Como enfatizar a parte prática?

Equipamentos



- Tem suas limitações, mas...
- Projeto de hardware
 - A maior parte é feita por simulações
 - Muito mais acessível, como veremos
 - Só a parte final depende do hardware
 - Conhecimento bastante transitório

EDA playground

Multiplex (structural, functional) x +

edaplayground.com/x/3PmP

EDA playground

Run Save Copy Aldec Riviera Pro 2019.10 is here! BTW: Mentor Precision examples: for VHDL and for (System)Verilog.

Brought to you by DOULOS

Languages & Libraries

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

None

Other Libraries

None

OVL 2.8.1

SVUnit 2.11

Enable TL-Verilog

Enable Easier UVM

Enable VUnit

Tools & Simulators

Aldec Riviera Pro 2019.10

Compile & Run Options

-timescale 1ns/1ns -sv2k9

+accosrv

Run Time: 10 ms

Use run.do Tcl file

Open EPWave after run

Download files after run

Examples

Community

Collaborate

Forum

Follow @edaplayground

testbench.sv values.tv

```
1 // Testbench
2 module test;
3
4 logic clk;
5 logic x1, x2, s;
6 logic f1, f2, f3, f_expected;
7
8 logic [31:0] vectornum, errors;
9 logic [103:0] testvectors[10000:0];
10
11 // Instantiate design under test
12 mux1 dut1(.x1(x1), .x2(x2), .s(s), .f(f1));
13 mux2 dut2(.x1(x1), .x2(x2), .s(s), .f(f2));
14 mux3 dut3(.x1(x1), .x2(x2), .s(s), .f(f3));
15
16 // generate clock
17 always begin
18     clk = 1; #50; clk = 0; #50;
19 end
20
21 // at start of test, load vectors
22 initial begin
23     $readmemh("values.tv", testvectors);
24     vectornum = 0; errors = 0;
25 end
26
27 // apply test vectors at rising edge of clock
28 always @(posedge clk)
29     begin
30         x1 = testvectors[vectornum][12];
31         x2 = testvectors[vectornum][8];
32         s = testvectors[vectornum][14];
```

design.sv

```
1 // Design
2
3 // Multiplex examples
4
5 // Structural
6 module mux1(
7     input x1, x2, s,
8     output f);
9     // optional declaration
10    wire k, g, h;
11    // inputs first, outputs after
12    not (k, s);
13    and (g, k, x1);
14    and (h, s, x2);
15    or (f, g, h);
16 endmodule
17
18 // Functional
19 module mux2(
20     input x1, x2, s,
21     output f);
22
23     assign f = (x1 & ~s) | (x2 & s);
24     assign f = s ? x2 : x1; // alternative
25 endmodule
26
27 // Behavioral
28 module mux3(
29     input x1, x2, s,
30     output reg f);
31     // declare output f as reg
32     always @(x1, x2, s)
```

Log Share

Multiplex (structural, functional & behavioral) 378 views and 3 likes Published (will appear in search results) Save

B I H " " " % " " "

Multiplex example (structural, functional & behavioral)

6

EDA playground - simuladores comerciais

The screenshot shows the EDA Playground interface. At the top, there's a browser address bar with the URL `edaplayground.com/x/3PmP`. Below the browser is a blue navigation bar with buttons for 'Run', 'Save', and 'Copy', and a status message: 'Aldec Riviera Pro 2019.10 is here! BTW: Mentor Precision examples: for VHDL and for (System)Verlog.'

The main content area is split into two panels: 'testbench.sv' and 'design.sv'. The 'testbench.sv' panel contains the following code:

```
1 // Testbench
2 module test;
3
4 logic clk;
5 logic x1, x2, s;
6 logic f1, f2, f3, f_expected;
7
8 logic [31:0] vectornum, errors;
9 logic [103:0] testvectors[10000:0];
10
11 // Instantiate design under test
12 mux1 dut1(.x1(x1), .x2(x2), .s(s), .f(f1));
13 mux2 dut2(.x1(x1), .x2(x2), .s(s), .f(f2));
14 mux3 dut3(.x1(x1), .x2(x2), .s(s), .f(f3));
15
16 // generate clock
17 always begin
18   clk = 1; #50; clk = 0; #50;
19 end
20
21 // at start of test, load vectors
22 initial begin
23   $readmemh("values.tv", testvectors);
24   vectornum = 0; errors = 0;
25 end
26
27 // apply test vectors at rising edge of clock
28 always @(posedge clk)
29   begin
30     x1 = testvectors[vectornum][12];
31     x2 = testvectors[vectornum][8];
32     s = testvectors[vectornum][4];
33   end
34 endmodule
```

The 'design.sv' panel contains the following code:

```
1 // Design
2
3 // Multiplex examples
4
5 // Structural
6 module mux1(
7   input x1, x2, s,
8   output f);
9 // optional declaration
10 wire k, g, h;
11 // inputs first, outputs after
12 not (k, s);
13 and (g, k, x1);
14 and (h, s, x2);
15 or (f, g, h);
16 endmodule
17
18 // Functional
19 module mux2(
20   input x1, x2, s,
21   output f);
22
23   assign f = (x1 & ~s) | (x2 & s);
24   assign f = s ? x2 : x1; // alternative
25 endmodule
26
27 // Behavioral
28 module mux3(
29   input x1, x2, s,
30   output reg f);
31 // declare output f as reg
32 always @(x1, x2, s)
33   f = (x1 & ~s) | (x2 & s);
34 endmodule
```

On the left side, there's a sidebar with 'Languages & Libraries' and 'Testbench + Design' sections. The 'Aldec Riviera Pro 2019.10' option is selected. Below the sidebar, there are social media sharing buttons (Twitter, Facebook, LinkedIn) and a 'Share' button. The share text is 'Multiplex (structural, functional & behavioral)'. There are also view and like counts: '378 views and 3 likes'. A 'Published' status is shown, and a 'Save' button is present.

At the bottom, there's a rich text editor with a toolbar and the text 'Multiplex example (structural, functional & behavioral)'.

EDA playground - resultado da simulação

Multiplex (structural, functions: x +

edaplayground.com/x/3PmP

EDA playground Run Save Copy Aldec Riviera Pro 2019.10 is here! BTW: Mentor Precision examples: for VHDL and for (System)Verilog. Playgrounds Profile

EPWave

From: 0ns To: 0ns

Get Signals Radix

clk
erron[31:0]
f1
f2
f3
f_expected
s
vectorname[31:0]
x1
x2

Note: To revert to EPWave opening in a new

Remoto

Suspensão das atividades presenciais

Disciplina: **Introdução à Programação Paralela (ACIEPE)**

Atividade Curricular de Integração Ensino, Pesquisa e Extensão
Oferecida para estudantes de diversos cursos e externos

- **8 semanas**
 - 60 horas
- **40 vagas**
 - 130+ inscr.
 - 40 defer.
 - 10 externos
- **Pré-requisito**
 - Prog. básica

Proposta:

- **plataformas**
 - CPUs
 - GPUs
 - FPGAs
- **proposta**
 - OpenMP
 - OpenCL
 - SYCL

Como?

- **Aulas Gravadas**
- **Questionários (10%)**
- **Fóruns (20%)**
- **Exercícios (30%)**
- **Projeto final (40%)**

```

$
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #include <omp.h>
5
6 #define DIM 1000 // ajuste a dimensão aqui...
7
8 float A [DIM] [DIM];
9 float B [DIM] [DIM];
10 float C [DIM] [DIM];
11
12 int
13 main(int argc, char *argv[])
14 {
15     int i, j, k;
16
17     for (i = 0; i < DIM; i++) {
18         for (j = 0; j < DIM; j++) {
19             A[i][j] = (float)rand() / (float)RAND_MAX / 5;
20             B[i][j] = (float)rand() / (float)RAND_MAX / 5;
21         }
22     }
23
24     srand(time(NULL));
25
26     float etime;
27     struct timespec inic, fim;
28     float time_seq;
29
30     for (int z = 1; z < 2 << 7; z <<= 1) {
31         clock_gettime(CLOCK_REALTIME, &inic);
32         omp_set_num_threads(z);
33         #pragma omp parallel for schedule(dynamic) private(j,k)
34         for (i = 0; i < DIM; i++)
35             for (j = 0; j < DIM; j++) {
36                 C[i][j] = 0;
37             }
38     }
39
40     clock_gettime(CLOCK_REALTIME, &fim);
41     time_seq = fim.tv_sec - inic.tv_sec + fim.tv_nsec - inic.tv_nsec / 1000000.0;
42
43     printf("2,1.039426,1.840570,0.920285\n");
44     printf("4,0.553771,3.454740,0.863685\n");
45     printf("8,0.543028,3.523092,0.440386\n");
46     printf("16,0.552371,3.463500,0.216469\n");
47     printf("32,0.577874,3.310645,0.103458\n");
48 }
49
real    0m5.223s
user    0m12.880s
sys     0m0.040s

```

Compile, link and run... 10 Share!



TECHIO BETA

Explore

Connect

Documentation



Snippets

My playgrounds



Introdução ao OpenCL



menotti

3,885 views



GitHub



Edit



2



0



Previous: Configuração do...

3/7 Olá mundo



Next: Um exemplo comple...



Olá mundo

Como de costume em qualquer tutorial de programação, vamos começar por este **olá mundo** adaptado de um livro gratuito disponível na internet. O programa não faz nenhum cálculo, apenas retorna a mensagem **"Hello, World!"** gerada no *kernel*. Como a entrada e saída padrão do sistema não podem ser usadas a partir do *kernel*, ele apenas vai preencher a cadeia de caracteres e enviá-la ou programa principal que irá imprimi-la na console.

Incluindo o arquivo de cabeçalho OpenCL

Toda aplicação OpenCL deve incluir o arquivo de cabeçalho `CL/cl.h`:

```
#include <CL/cl.h>
```

Mas antes disso, nós estamos definindo

`CL_USE_DEPRECATED_OPENCL_1_2_APIS` para usar a versão 1.2 da API sem gerar *warnings* durante a compilação:

hello_world.cpp

hello.cl



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define CL_USE_DEPRECATED_OPENCL_1_2_APIS
5
6 #include <CL/cl.h>
7
8 #define MEM_SIZE (128)
9 #define MAX_SOURCE_SIZE (0x100000)
10
11 int main()
12 {
13     cl_device_id device_id = NULL;
14     cl_context context = NULL;
15     cl_command_queue command_queue = NULL;
16     cl_mem memobj = NULL;
17     cl_program program = NULL;
18     cl_kernel kernel = NULL;
19     cl_platform_id platform_id = NULL;
20     cl_uint ret_num_devices;
21     cl_uint ret_num_platforms;
22     cl_int ret;
23
24     char string[MEM_SIZE];
25
26     FILE *fp;
27     char fileName[] = "../kernels/hello.cl";
28     char *source_str;
29     size_t source_size;
30
```

Introdução ao OpenCL

menotti | 3,885 views | GitHub | Edit | 🇧🇷

♥️ 2 💬 0 🔗 ⏪ Previous: Configuração do... 3/7 Olá mundo 📄 Next: Um exemplo comple... 🔗

Vamos executar!

🗑️ Hello World - OpenCL

Run

⏪ Previous: Configuração... Next: Um exemplo com... ⏩



🔗 Suggested playgrounds

🏠 Aprender le C++

By 👤 users

👁️ 6.078 ❤️ 100 🗨️ 2

```
hello_world.cpp | hello.cl | ☰
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define CL_USE_DEPRECATED_OPENCL_1_2_APIS
5
6 #include <CL/cl.h>
7
8 #define MEM_SIZE (128)
9 #define MAX_SOURCE_SIZE (0x100000)
10
11 int main()
12 {
13     cl_device_id device_id = NULL;
14     cl_context context = NULL;
15     cl_command_queue command_queue = NULL;
16     cl_mem memobj = NULL;
17     cl_program program = NULL;
18     cl_kernel kernel = NULL;
19     cl_platform_id platform_id = NULL;
20     cl_uint ret_num_devices;
21     cl_uint ret_num_platforms;
22     cl_int ret;
23
24     char string[MEM_SIZE];
25
26     FILE *fp;
27     char fileName[] = "./kernels/hello.cl";
28     char *source_str;
29     size_t source_size;
30
```

Introdução ao OpenCL

menotti | 3,885 views | GitHub | Edit | 🇧🇷

♥️ 2 💬 0 🔗

⏪ Previous: Configuração do... 3/7 Olá mundo ⏩ Next: Um exemplo comple... 🔗

Vamos executar!

🔗 Hello World - OpenCL

🔄 In progress 🛑 Stop

📄 Standard Output

Hello, World!

⏪ Previous: Configuração... Next: Um exemplo com... ⏩



```
hello_world.cpp | hello.cl
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define CL_USE_DEPRECATED_OPENCL_1_2_APIS
5
6 #include <CL/cl.h>
7
8 #define MEM_SIZE (128)
9 #define MAX_SOURCE_SIZE (0x100000)
10
11 int main()
12 {
13     cl_device_id device_id = NULL;
14     cl_context context = NULL;
15     cl_command_queue command_queue = NULL;
16     cl_mem memobj = NULL;
17     cl_program program = NULL;
18     cl_kernel kernel = NULL;
19     cl_platform_id platform_id = NULL;
20     cl_uint ret_num_devices;
21     cl_uint ret_num_platforms;
22     cl_int ret;
23
24     char string[MEM_SIZE];
25
26     FILE *fp;
27     char fileName[] = "./kernels/hello.cl";
28     char *source_str;
29     size_t source_size;
30
```



Introdução ao OpenCL

menotti | 3,885 views | GitHub | Edit |

2 0 [Previous: Configuração do...](#) 3/7 Olá mundo [Next: Um exemplo comple...](#)

Vamos executar!

Hello World - OpenCL

Success!

Standard Output

Hello, World!

[Previous: Configuração...](#) [Next: Um exemplo com...](#)



```
hello_world.cpp | hello.cl | 
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define CL_USE_DEPRECATED_OPENCL_1_2_APIS
5
6 #include <CL/cl.h>
7
8 #define MEM_SIZE (128)
9 #define MAX_SOURCE_SIZE (0x100000)
10
11 int main()
12 {
13     cl_device_id device_id = NULL;
14     cl_context context = NULL;
15     cl_command_queue command_queue = NULL;
16     cl_mem memobj = NULL;
17     cl_program program = NULL;
18     cl_kernel kernel = NULL;
19     cl_platform_id platform_id = NULL;
20     cl_uint ret_num_devices;
21     cl_uint ret_num_platforms;
22     cl_int ret;
23
24     char string[MEM_SIZE];
25
26     FILE *fp;
27     char fileName[] = "./kernels/hello.cl";
28     char *source_str;
29     size_t source_size;
30
```

Chrome File Edit View History Bookmarks People Tab Window Help

WhatsApp x SERCET - On x Curso: LÓGIC x Colfax SEE | x Home | Colfa x Dashboard | x cadastro na x 2020_Coben x https://mail- x +

https://access.colfaxresearch.com/?uuld=voce-nao-pode-ver-isso

Colfax Cluster for Coursera **Home** Learn Connect Program Compute Log out

Welcome to the Colfax Cluster for Coursera!

Learn



what to expect on the Colfax Cluster for Coursera

Connect



from your home computer to the cloud

Program



using modern code practices

Compute



with cluster job management tools



To: Coursera Learner 25452
From: Colfax Cluster for Coursera

Dear Coursera Learner 25452,

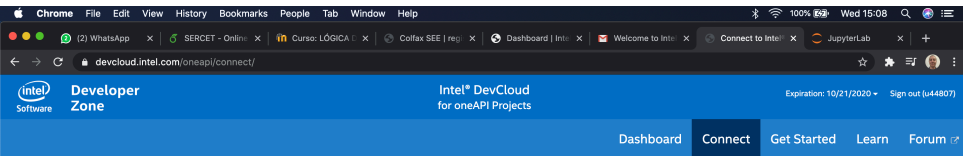
Please browse this portal for connection and usage instructions. If you have any problems connecting to or using the Colfax Cluster for Coursera, please search or post on the [Forum](#). Include your user ID in your post: u47044@c008.

Your account is active until **Sep 11 2020 15:12:57 UTC**. Your account and data will be deleted at expiration, so transfer out any data you wish to preserve before this date.

By using the Colfax Cluster for Coursera, you agree to abide by the following terms:

- [Colfax Terms of Service](#)

Sincerely,
Colfax Cluster for Coursera



intel Software Developer Zone Intel® DevCloud for oneAPI Projects Expiration: 10/21/2020 - Sign out (u44807)

Dashboard Connect Get Started Learn Forum

CONNECT TO INTEL® DEVCLOUD

Use a Secure Shell (SSH) client terminal or a JupyterLab* in your web browser to begin.



Connect with a Terminal

Select your operating system to get started.

[Windows* with Cygwin](#) (preferred)
[Windows* with PuTTY](#) (legacy)
[Linux* or macOS](#) (SSH client)
[Visual Studio Code*](#) (VS Code*)



Connect with JupyterLab*

[One-click Log In](#)

or

URL: <https://jupyter.oneapi.devcloud.intel.com/>
Username: u44807
Password: [show](#)

```
u44807@login-2: ~/oil_gas_fpga/apps/CMP/SYCL/src ℃#1  
  
auto a_c      = b_c.get_access<sycl::access::mode::read_write>(cgh);  
auto a_h      = b_h.get_access<sycl::access::mode::read_write>(cgh);  
auto a_num     = b_num.get_access<sycl::access::mode::read_write>(cgh);  
auto a_stt     = b_stt.get_access<sycl::access::mode::read_write>(cgh);  
  
cgh.parallel_for<class kernelC>(sycl::range<1>(ns*nc), [=](sycl::id<1> i) {  
  
    real _den = 0.0f, _ac_linear = 0.0f, _ac_squared = 0.0f;  
    real _num[MAX_W], m = 0.0f;  
    int err = 0;  
  
    if(i < ns*nc)  
    {  
        int c_id = i % nc;  
        int t0 = i / nc;  
  
        real _c = a_c[c_id];  
        real _t0 = _dt * t0;  
        _t0 = _t0 * _t0;  
  
        // start _num with zeros  
        for(int j=0; j < _w; j++) _num[j] = 0.0f;  
  
        for(int t_id=t_id0; t_id < t_idf; t_id++) {  
            // Evaluate t  
            real t = sycl::sqrt(_t0 + _c * a_h[t_id]);  
  
            int it = (int)( t * _idt );  
            int ittau = it - _tau;  
            real x = t * _idt - (real)it;  
  
            if(ittau >= 0 && it + _tau + 1 < ns) {
```

Old School

```
u44807@login-2: ~/oil_gas_fpga/apps/CMP/SYCL/bin
u44807@login-2:~/oil_gas_fpga/apps/CMP/SYCL$ ls
CMakeLists.txt  README.md  include  src
u44807@login-2:~/oil_gas_fpga/apps/CMP/SYCL$ mkdir bin
u44807@login-2:~/oil_gas_fpga/apps/CMP/SYCL$ cd bin
u44807@login-2:~/oil_gas_fpga/apps/CMP/SYCL/bin$ cmake ..
-- The C compiler identification is GNU 7.4.0
-- The CXX compiler identification is GNU 7.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/u44807/oil_gas_fpga/apps/CMP/SYCL/bin
u44807@login-2:~/oil_gas_fpga/apps/CMP/SYCL/bin$ make
Scanning dependencies of target cmp-sycl2
[ 14%] Building CXX object CMakeFiles/cmp-sycl2.dir/src/log.cpp.o
[ 28%] Building CXX object CMakeFiles/cmp-sycl2.dir/src/su_trace.cpp.o
[ 42%] Building CXX object CMakeFiles/cmp-sycl2.dir/src/su_gather.cpp.o
[ 57%] Building CXX object CMakeFiles/cmp-sycl2.dir/src/su_cdp.cpp.o
[ 71%] Building CXX object CMakeFiles/cmp-sycl2.dir/src/parser.cpp.o
[ 85%] Building CXX object CMakeFiles/cmp-sycl2.dir/src/main.cpp.o
[100%] Linking CXX executable cmp-sycl2
[100%] Built target cmp-sycl2
u44807@login-2:~/oil_gas_fpga/apps/CMP/SYCL/bin$
```

Jupyter Notebooks

The screenshot shows a Jupyter Notebook running in a browser. The notebook contains a Python script that reads a CSV file, groups data by platform, and plots the average execution time in milliseconds. The plot shows three bars representing different platforms: Intel(R) Core(TM) i5-7360U CPU @ 2.30GHz, Intel(R) FPGA Emulation Device, and Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz. The execution times are approximately 33 ms, 7 ms, and 6 ms respectively.

```
[9]: # Plot results
df = pd.read_csv(FILENAME)
by_version = df[df.version == 'OpenCL'].groupby(by="platform", sort=False)
avg_time = by_version.mean()
std = by_version.std()
plt = avg_time.plot(legend=False, kind="bar", yerr=std)
plt.set_title("Average execution time (OpenCL)")
plt.set_ylabel("Execution time (ms)")
```

[9]: Text(0, 0.5, 'Execution time (ms)')

Average execution time (OpenCL)

platform	Execution time (ms)
Intel(R) Core(TM) i5-7360U CPU @ 2.30GHz	~33
Intel(R) FPGA Emulation Device	~7
Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz	~6

Execution time (ms)

platform

Intel(R) Core(TM) i5-7360U CPU @ 2.30GHz

Intel(R) FPGA Emulation Device

Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz

0 4 No Kernel | Mode: Command Ln 8, Col 38 27stencil.ipynb

Escolha a resposta

Resposta numérica: 1 (Anônimo)

(geral) Qual a sua avaliação geral sobre o curso?*

Achei bem **dinâmico e diversificado**. Gostei da organização apresentada, pois facilitou muito o acompanhamento da matéria. Senti um pouco de dificuldade em alguns momentos, o que por um lado mostra que o curso não apresenta assuntos banais.

(melhorar) O que você acha que poderia melhorar para as próximas ofertas?

Talvez um maior esclarecimento dos "pré-requisitos", por assim dizer. Acredito que esses não estavam especificados na oferta da ACIEPE, mas isso não se apresentou como um grande problema...

(manter) O que você acha que poderia ser destacado como o melhor aspecto do curso?

A diversidade da exploração de assuntos desconhecidos (por mim, pelo menos). Gostei muito da **abrangência** do curso.

Feedback

Escolha a resposta

Resposta numérica: 2 (Anônimo)

(geral) Qual a sua avaliação geral sobre o curso?*

Eu gostei muito da ACIEPE. Mesmo não sendo da área de ciência da computação foi possível acompanhar as aulas e realizar as atividades. Também gostei da variedade das atividades avaliativas e da proposta mais "aberta" do trabalho final, que me permitiu fazer uma conexão com minha área de estudos. O curso foi muito bom.

(melhorar) O que você acha que poderia melhorar para as próximas ofertas?

Se fosse possível disponibilizar um tutorial rápido de instalação de certas ferramentas (levando em conta que nem todos os matriculados são do curso de ciência da computação) seria muito bom. Eu por exemplo, tentei utilizar SYCL na minha máquina com o Codeplay, mas não consegui por não entender muito bem o tutorial deles. Mas reconheço que provavelmente os alunos da área não tiveram esse tipo de problema. Além disso, não foi um impeditivo já que foram disponibilizadas outras plataformas pra realizar os exercícios. Mas seria interessante, rodar no próprio dispositivo, (como pude fazer por exemplo com OpenCL).

(manter) O que você acha que poderia ser destacado como o melhor aspecto do curso?

Cito dois: a variedade de ferramentas apresentadas e a variedade de formas de avaliação.

Feedback

Escolha a resposta

Resposta numérica: 3 (Anônimo)

(geral) Qual a sua avaliação geral sobre o curso?*

Eu achei o tema do curso muito interessante e o métodos de paralelização abordado, porém acho que faltou um pouco mais de **contato entre aluno e professor.**

(melhorar) O que você acha que poderia melhorar para as próximas ofertas?

Acho que a realização de **encontros síncronos pelo menos uma vez por semana** para tirar dúvidas.

(manter) O que você acha que poderia ser destacado como o melhor aspecto do curso?

O conteúdo foi muito bom, a **aulas gravadas no youtube ajudou muito para evitar conflito de horário** entre outras coisas

Escolha a resposta

Resposta numérica: 4 (Anônimo)

(geral) Qual a sua avaliação geral sobre o curso?*

Eu gostei bastante do curso, principalmente em ter esse contato com uma área que é pouca abordada durante a graduação. Contudo, tive bastante dificuldade na metade final, pois foram ensinadas muitas tecnologias de uma semana para a outra.

(melhorar) O que você acha que poderia melhorar para as próximas ofertas?

Creio que seria melhor aprofundar em uma das tecnologias (OpenMP, OpenCL ou SYCL) em vez de abordar as três.

(manter) O que você acha que poderia ser destacado como o melhor aspecto do curso?

É um curso que proporciona um panorama bastante amplo da área.

Feedback

Escolha a resposta

Resposta numérica: 5 (Anônimo)

(geral) Qual a sua avaliação geral sobre o curso?*

Foi um curso bem legal, deu pra **conhecer muita coisa sobre programação paralela** (eu por exemplo não conhecia nada sobre o SYCL e inclusive confundia o OPENMP com o OPENCL) porém, senti dificuldades na hora de fazer as análises de desempenho, não sei se é uma dificuldade que vem no meu embasamento sobre hardware mas as vezes eu **não conseguia correlacionar bem as coisas.**

(melhorar) O que você acha que poderia melhorar para as próximas ofertas?

Mesmo que seja introdutorio, acredito que seja legal **focar apenas em uma linguagem**, como por exemplo o OPENMP ou OPENCL (esse pq sei que muita gente preferiu por causa do PYTHON). Além disso, talvez seja legal focar nessa parte de análise de desempenho, dando exemplos do que acontece no hardware quando vc trabalha com workgroups de tamanhos diferentes e etc.

Talvez isso tenha ficado **muito abstrato.**

(manter) O que você acha que poderia ser destacado como o melhor aspecto do curso?

Na minha opinião, são duas coisas: A **prática com as linguagens apresentadas** (por isso, retomando o que eu respondi anteriormente, talvez fosse melhor focar apenas em uma) e a **interatividade nos fóruns.**

Escolha a resposta

Resposta numérica: 6 (Anônimo)

(geral) Qual a sua avaliação geral sobre o curso?*

Por se tratar de uma ACIEPE, **atendeu as expectativas, dando uma boa abordagem introdutória** à programação paralela. Aos que tiverem interesse em continuar estudando sobre o assunto, tenho plena convicção que tiveram uma noção dos princípios e desafios da área.

(melhorar) O que você acha que poderia melhorar para as próximas ofertas?

Talvez **mais exercícios práticos e abordagem em outras linguagens de programação.**

(manter) O que você acha que poderia ser destacado como o melhor aspecto do curso?

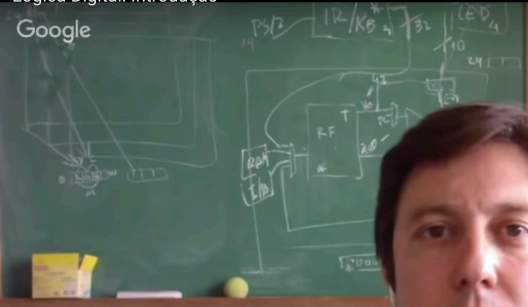
A **organização** do curso e a **qualidade do material disponibilizado pelos professores.**

Galeria

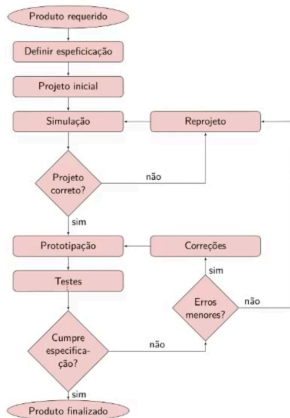
Youtuber principiante

Lógica Digital: Introdução

Google

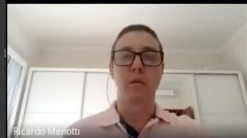


Hardware digital



Procedimentos de Avaliação

- ▶ A avaliação será composta pela média ponderada das atividades a seguir com os respectivos pesos:
 - ▶ 10% - fóruns de discussão;
 - ▶ 20% - questionários;
 - ▶ 30% - exercícios de programação;
 - ▶ 40% - projeto final;
- ▶ Será considerado aprovado o participante que obtiver média final igual ou superior a 6,0 e pelo menos 75% de frequência, que será apurada por meio do envio das atividades propostas em cada uma das 8 semanas do curso.



Chroma Key

Live@DC: Gravação de videoaulas

The screenshot displays the OBS Studio interface during a live recording session. The main preview window at the top center shows a pyramid-shaped arrangement of video sources, with the largest source at the bottom and smaller ones stacked on top. Below the preview, the interface is divided into several panels: 'Scenes' (left), 'Audio Mixer' (center-left), 'Sources' (center-right), 'Scene Transitions' (right), and 'Controls' (far right). The 'Audio Mixer' panel shows two microphone sources, 'Mic/Aux' at -6.8 dB and 'Mic/Aux 2' at -10.3 dB. The 'Sources' panel lists 'Video Capture' and 'Display Capture'. The 'Scene Transitions' panel is set to 'Fade' with a duration of 300 ms. The 'Controls' panel includes buttons for 'Stop Streaming' and 'Start Recording'. At the bottom of the interface, a status bar shows 'Dropped Frames 0 (0.0%)', 'LIVE: 00:22:52', 'REC: 00:00:00', and 'CPU:'. A video player overlay is visible at the very bottom of the image, showing a progress bar at 9:27 / 57:29 and standard playback controls.

9:27 / 57:29



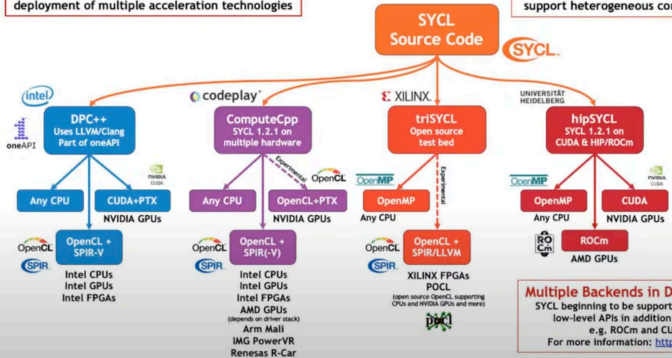
Slides (16:9) + Chroma Key

Programacao Paralela: SYCL

Implementações

SYCL, OpenCL and SPIR-V, as open industry standards, enable flexible integration and deployment of multiple acceleration technologies

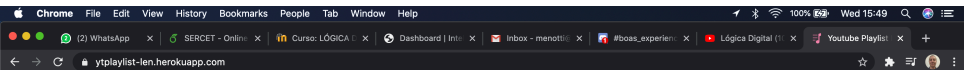
SYCL enables Khronos to influence ISO C++ to (eventually) support heterogeneous compute



Multiple Backends in Development
SYCL beginning to be supported on multiple low-level APIs in addition to OpenCL, e.g. ROCm and CUDA
For more information: <http://sycl.cpp>



YouTube Playlist Length



Find the length of any YouTube playlist :

Get Length

You can enter a playlist link, playlist ID or even a video link from the playlist!
This only works with playlists with upto 500 videos. Need more than that? [Check here.](#)

No of videos : 33

Average length of video : 18 minutes, 10 seconds

Total length of playlist : 9 hours, 59 minutes, 46 seconds

At 1.25x : 7 hours, 59 minutes, 48 seconds

At 1.50x : 6 hours, 39 minutes, 50 seconds

At 1.75x : 5 hours, 42 minutes, 43 seconds

At 2.00x : 4 hours, 59 minutes, 53 seconds

Need some 🍺 beer money? 😊

- 🔥 Brave Browser allows you to view privacy-respecting ads and refer others to earn 💰 money.
- 🍷 Consider supporting me by [using my affiliate link to install Brave browser](#) or [donating on Paypal](#) .

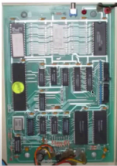
Link para trechos do vídeo no YouTube

YouTube BR Search

Google

Standard Chips

- ▶ Disponíveis para realizar funções comuns;
- ▶ Agrupados e conectados para construir um circuito;
- ▶ Muito usados até a década de 80, mas consomem muito espaço na placa;
- ▶ Possuem funcionalidades fixas, ou seja, não podem ser mudadas após a fabricação.




4:28 / 11:34

Lógica Digital: Introdução

866 views

LIKE DISLIKE SHARE SAVE ...

 Ricardo Menotti [ANALYTICS](#) [EDIT VIDEO](#)

Neste vídeo faço uma introdução sobre o curso de Lógica Digital usando Verilog, nele comento sobre algumas tecnologias e sobre o fluxo de projeto que usaremos no curso.

- [1:15 - Hardware Digital](#)
- [3:26 - Lei de Moore](#)
- [4:28 - Standard Chips](#)
- [5:44 - Field-Programmable Gate Array \(FPGA\)](#)
- [6:57 - Application-Specific Integrated Circuit \(ASIC\)](#)
- [8:45 - Fluxo de projeto de Hardware Digital](#)
- [11:00 - Referências](#)

SHOW LESS





Das metodologias ativas ao ensino remoto: ferramentas e experiências

Ensino prático a distância na computação

Prof. Ricardo Menotti <menotti@ufscar.br>

13 de agosto de 2020

SEMANA DO ENSINO REMOTO PARA AS CIÊNCIAS EXATAS E TECNOLÓGICAS

Departamento de Computação

Centro de Ciências Exatas e de Tecnologia

Universidade Federal de São Carlos

Bloom's Taxonomy Verbs

Evaluation

Make and defend judgments based on internal evidence or external criteria.

appraise
argue assess attach
choose compare conclude
contrast defend describe discriminate
estimate evaluate explain judge justify interpret
relate predict rate select summarize support value

Synthesis

Compile component ideas into a new whole or propose alternative solutions.

arrange assemble categorize collect combine comply
compose construct create design develop devise explain
formulate generate plan prepare rearrange reconstruct relate
reorganize revise rewrite set up summarize synthesize tell write

Analysis

Break down objects or ideas into simpler parts and find evidence to support generalizations.

analyze appraise breakdown calculate categorize compare
contrast criticize diagram differentiate discriminate distinguish
examine experiment identify illustrate infer model outline
point out question relate select separate subdivide test

Application

Apply knowledge to actual situations.

apply change choose compute demonstrate discover
dramatize employ illustrate interpret manipulate
modify operate practice predict prepare produce
relate schedule show sketch solve use write

Comprehension

Demonstrate an understanding of the facts.

classify convert defend describe discuss
distinguish estimate explain express
extend generalized give example(s)
identify indicate infer locate paraphrase
predict recognize rewrite review select
summarize translate

Knowledge

Remember previously learned information.

arrange define describe duplicate
identify label list match memorize
name order outline recognize
relate recall repeat reproduce
select state

Higher Order
Thinking Skills

Lower Order
Thinking Skills

By Fracus Learning, CC BY-SA 4.0,

<https://commons.wikimedia.org/w/index.php?curid=69357265>