





**THE IMAGINATION UNIVERSITY PROGRAMME**

# **Guia do Seminário RVfpga (Workshop)**

## Agradecimientos



Imagination  
university programme

Imagination

**AUTHORS**  
Prof. Sarah Harris  
Prof. Daniel Chaver  
Zubair Kakakhel  
M. Hamza Liaqat

**ADVISER**  
Prof. David Patterson

**CONTRIBUTORS**  
Robert Owen  
Olof Kindgren  
Prof. Luis Piñuel  
Ivan Kravets  
Valerii Koval  
Ted Marena  
Prof. Roy Kravitz

**ASSOCIATES**  
Prof. José Ignacio Gómez  
Prof. Christian Tenllado  
Prof. Daniel León  
Prof. Katzalin Olcoz  
Prof. Alberto del Barrio  
Prof. Fernando Castro  
Prof. Manuel Prieto

Prof. Francisco Tirado  
Prof. Román Hermida  
Prof. Julio Villalba  
Prof. Ataur Patwary  
Cathal McCabe  
Dan Hugo  
Braden Harwood  
Prof. David Burnett

Gage Elerding  
Prof. Brian Cruickshank  
Deepen Parmar  
Thong Doan  
Oliver Rew  
Niko Nikolay  
Guanyang He  
Prof. Peng Liu

## Sponsors and Supporters



### AUTORES

- Prof. Sarah Harris (<https://www.linkedin.com/in/sarah-harris-12720697/>)
- Prof. Daniel Chaver (<https://www.linkedin.com/in/daniel-chaver-a5056a156/>)
- Zubair Kakakhel (<https://www.linkedin.com/in/zubairk/>)
- M. Hamza Liaqat (<https://www.linkedin.com/in/muhammad-hamza-liaqat-ab73a0195/>)

### CONSULTOR

- Prof. David Patterson (<https://www.linkedin.com/in/dave-patterson-408225/>)

### COLABORADORES

- Robert Owen (<https://www.linkedin.com/in/robert-owen-4335931/>)
- Olof Kindgren (<https://www.linkedin.com/in/olofkindgren/>)
- Prof. Luis Piñuel (<https://www.linkedin.com/in/lpinuel/>)
- Ivan Kravets (<https://www.linkedin.com/in/ivankravets/>)
- Valerii Koval (<https://www.linkedin.com/in/valeros/>)
- Ted Marena (<https://www.linkedin.com/in/tedmarena/>)
- Prof. Roy Kravitz (<https://www.linkedin.com/in/roy-kravitz-4725963/>)

### ASOCIADOS

- Prof. José Ignacio Gómez (<https://www.linkedin.com/in/jos%C3%A9-ignacio-gomez-182b981/>)
- Prof. Christian Tenllado (<https://www.linkedin.com/in/christian-tenllado-31578659/>)
- Prof. Daniel León (<https://www.linkedin.com/in/danileon-ufv/>)
- Prof. Katzalin Olcoz (<https://www.linkedin.com/in/katzalin-olcoz-herrero-5724b0200/>)
- Prof. Alberto del Barrio (<https://www.linkedin.com/in/alberto-antonio-del-barrio-garc%C3%ADa-1a85586a/>)
- Prof. Fernando Castro (<https://www.linkedin.com/in/fernando-castro-5993103a/>)
- Prof. Manuel Prieto (<https://www.linkedin.com/in/manuel-prieto-matias-02470b8b/>)
- Prof. Francisco Tirado (<https://www.linkedin.com/in/francisco-tirado-fern%C3%A1ndez-40a45570/>)
- Prof. Román Hermida (<https://www.linkedin.com/in/roman-hermida-correa-a4175645/>)
- Prof. Julio Villalba (<https://www.linkedin.com/in/julio-villalba-moreno-97474824/>)
- Cathal McCabe (<https://www.linkedin.com/in/cathalmccabe/>)
- Dan Hugo (<https://www.linkedin.com/in/danhugo/>)
- Braden Harwood (<https://www.linkedin.com/in/braden-harwood/>)
- David Burnett (<https://www.linkedin.com/in/david-burnett-3b03778/>)
- Gage Elerding (<https://www.linkedin.com/in/gage-elerding-052b16106/>)
- Brian Cruickshank (<https://www.linkedin.com/in/bcruiksh/>)
- Deepen Parmar (<https://www.linkedin.com/in/deepen-parmar/>)
- Thong Doan (<https://www.linkedin.com/in/thong-doan/>)
- Oliver Rew (<https://www.linkedin.com/in/oliver-rew/>)
- Niko Nikolay (<https://www.linkedin.com/in/roy-kravitz-4725963/>)
- Guanyang He (<https://www.linkedin.com/in/quanyang-he-5775ba109/>)
- Prof. Ataur Patwary (<https://www.linkedin.com/in/ataurpatwary/>)
- Prof. Peng Liu (<https://person.zju.edu.cn/liupeng>)

*“O RISC-V está a transformar o projecto do processador e o co-projecto de software/hardware. O RISC-V é uma arquitectura aberta, que permite implementações de hardware em código aberto. Esta nova opção significa que o desenvolvimento de software pode ocorrer a par do desenvolvimento de hardware, acelerando o caminho do design. O curso RVfpga melhora a compreensão não só dos processadores RISC-V, mas também do ecossistema RISC-V e dos SoCs RISC-V. Este curso proporciona uma compreensão profunda de uma arquitectura de processadores de robustez industrial e de um sistema de popularidade crescente, que se revelará útil ao longo das suas carreiras académicas e industriais.”*

– David Patterson, Universidade da Califórnia, Berkeley

## 1. RESUMO

Este Guia de Seminário RVfpga fornece uma orientação para um seminário de um dia sobre o RVfpga. Os participantes podem seguir este guia em paralelo com o seminário, e os instrutores podem utilizá-lo para orientar o seminário. Este guia inclui o seguinte:

- **Resumo** dos materiais do seminário que lista os tópicos e os números dos slides associados (de RVfpga\_Slides.pptx), demonstrações e actividades práticas. Um instrutor pode optar por incluir apenas um subconjunto das actividades práticas e de demonstração (demo).
- **Sugestão de Calendarização do Workshop** para um seminário de um dia. Note-se que, embora o seminário esteja organizado para durar um dia, algumas das demonstrações e aspectos práticos podem ser removidos para o converter num seminário de meio dia ou mesmo num seminário de várias horas, conforme necessário.
- **Demonstrações e Detalhes Práticos** sobre cada tópico do esboço que fornece informações adicionais para completar as demonstrações e actividades práticas.

Para tirar o máximo partido do seminário, os participantes devem solicitar o Pacote RVfpga à Imagination Technologies (<https://university.imgtec.com/rvfpga/>) antes de participarem no Seminário RVfpga. Este Guia de seminário está incluído como parte do pacote RVfpga na pasta RVfpga\Documents. Os instrutores do seminário podem optar por fornecer este guia aos participantes como material impresso, para que eles possam acompanhar facilmente durante o seminário.

### Conteúdo do Guia do Seminário RVfpga:

1. Resumo.....	pag. 4
2. Programa do Seminário.....	pag. 5
3. Sugestão de Calendarização do Workshop.....	pag. 7
4. Demonstrações e Detalhes Práticos.....	pag. 8

## 2. Programa do Seminário

### Part 0: Introdução, instalação, programação e simulação

#	Tópico	Slides	Demonstração	Prática
<b>Introdução e instalação</b>				
A	Introdução e visão global	1-31		
B	Instalação mínima do RVfpga	33-34		GSG Secção 2.A. Instalar VSCode, PlatformIO & GTKWave
C	Instalar a plataforma Chips Alliance e executar um programa de exemplo na placa	36		GSG Secção 2.B. Utilizar PlatformIO para executar LEDswitches na placa FPGA
<b>Programação e Simulação</b>				
D	AL_Operations: Executar e depurar na placa	37		GSG Secção 6.B: seguir as instruções para executar/depurar o programa na placa: AL_Operações
E	AL_Operations: Simular no Whisper	38		Secção 8 do GSG: seguir as instruções (saltar o passo 5 - fá-lo-á mais tarde)
F	AL_Operations: Simular no Verilator	39		GSG Secção 7: seguir as instruções (saltar a geração de binários de simulação)
G	HelloWorld: Executar na placa	40		Secção 6.F do GSG: seguir as instruções
H	HelloWorld: Simular no Whisper	41		GSG Secção 8: seguir as instruções (incluir o passo 5)

## Parte 1: Labs RVfpga– Parte 1 – Labs 1 a 10: Programação e E/S

#	Tópico	Slides	Demonstração	Prática
<b>Labs RVfpga – Parte 1</b>				
1	Lab 1: Programação em C	42-49	Exercício 4 & 9	
2	Lab 2: Programação em Assembly	50-56		Criar um projecto PlatformIO (seguir a secção 3)
3	Lab 3: Chamadas a funções	57-70	Exercício 6	
4	Lab 4: Processamento de imagens - C & Assembly	71-70	Exemplo: Transformação de imagem	
5	Lab 5: Projecto Vivado	78-79	Criar um projeto Vivado - Secção 2	
6	Lab 6: Introdução às E/S	80-91		Exercício 5
7	Lab 7: Mostradores de 7 segmentos	92-101		Exercício 1
8	Lab 8: Temporizadores	102-110	Exercícios 1 & 3	
9	Lab 9: E/S com interrupções	111-119	Exemplo – Secção 5: 1. LED-Switch_C-Lang 2. 7SegDispl_C-Lang 3. 7SegDispl_Interrupts_C-Lang	Exercício 1
10	Lab 10: Barramentos série	120-130	Exercícios 1 & 3	

## Parte 2: Labs RVfpga– Part 2 – Labs 11 a 20: Núcleo e sistema de memória

#	Tópico	Slides	Demonstração	Prática
<b>Labs RVfpga– Parte 2</b>				
11	Lab 11: Configuração do SweRV EH1 e monitorização do desempenho	134-152		Secção 2.D – Figura 11 Secção 3.B – Figura 13
12	Lab 12: <code>add</code>	153-161	Exercício 7	
13	Lab 13: <code>lw</code> e <code>sw</code>	162-179		
14	Lab 14: Conflitos estruturais	180-190		Tarefa (na Secção 2.A: <code>MUL_Instruction</code> )
15	Lab 15: Conflitos de dados	191-208	Tasks (nas Secções 2.A & 3)	
16	Lab16: Conflitos de controlo e saltos	209-220		Tarefa (na Secção 3)
17	Lab 17: Execução Superscalar	221-235	Exercícios 3, 5 e 6	
18	Lab 18: Adição de novas funcionalidades	236-237		Exercícios 1, 3 e 4
19	Lab 19: I\$	238-255		
20	Lab 20: ICCM, DCCM & Avaliação de desempenho	256-270		Secções 3.A, 3.B e 3.C

### 3. Sugestão de Calendarização do Workshop

Hora	Tópico #	Tópico
09:00-09:30	A	Introdução ao RVfpga
09:30-10:00	B-C	Instalação de ferramentas RVfpga
10:00-10:30	D-H	Programação e Simulação
10:30-11:00	1-4	Labs 1-4: Programação em C e Assembly
11:00-11:15		<i>Intervalo</i>
11:15-11:30	5	Lab 5: Projeto RVfpga no Vivado
11:30-12:30	6-10	Labs 6-10: E/S do RVfpga
12:30-13:15		<i>Almoço</i>
13:15-14:45	11-16	Labs 11-16: Configuração do núcleo e pipeline
14:45-15:00		<i>Intervalo</i>
15:00-16:00	17-20	Labs 17-20: Superescalar, Memória e Avaliação
16:00-17:00		Conclusões e Questões

## 4. Demonstrações e Detalhes Práticos

### PARTE 0. Introdução, instalação, programação e ferramentas

#### A. Introdução

Não há demonstração nem acção prática.

#### B. Instalação mínima do RVfpga

##### PRÁTICA:

##### 1. Instalar o VSCode e o PlatformIO:

Siga as instruções da Secção 2.A do Guia de Iniciação - *Getting Started Guide* (“*Minimal installation: VSCode, PlatformIO and Nexys A7 board drivers*”).

- a. Instalar o VSCode
- b. Instalar o PlatformIO dentro do VSCode
- c. Instalar os *drivers* para a Nexys A7:
  - i. No Linux, siga as instruções no final desta secção.
  - ii. No Windows, seguir as instruções do Apêndice B das GSG.
  - iii. No MacOS isto não é necessário.

##### 2. Instalar o GTKWave:

- a. **Linux:** Abra um terminal e instale o GTKWave:
  - `sudo apt-get install -y gtkwave`
- b. **Windows:** O GTKWave pode ser descarregado como um pacote pré-compilado a partir de <https://sourceforge.net/projects/gtkwave/files/>. Procure o pacote mais recente do Windows, descarregue-o, execute-o e utilize-o no seu computador Windows.
- c. **MacOS:** Use o Homebrew para instalar o gtkwave. Mas desta vez é necessário usar o *cask* porque é uma aplicação GUI do macOS. Escreva os seguintes comandos numa janela de Terminal:

- `brew tap homebrew/cask`
- `brew cask install xquartz`
- `brew cask install gtkwave`

Após a instalação, um ícone para *gtkwave.app* deve aparecer na pasta Application. Para o utilizar a partir da linha de comandos, poderá ser necessário instalar o módulo Switch do Perl:

- `cpan install Switch`

## C. Instalar a plataforma ChipsAlliance e executar o programa de exemplo (LEDsSwitches) na placa FPGA Nexys-A7-100T

**PRÁTICA:** Siga as instruções da Secção 2.B do Guia de Iniciação (“*Download RVfpgaNexys onto FPGA and run programs on RVfpgaNexys*”), conforme resumido a seguir:

- Passo 1. Ligar a placa Nexys A7 FPGA ao computador e ligar a placa
- Passo 2. Abra o PlatformIO e o programa em C
- Passo 3. Configure o RVfpgaNexys na placa Nexys A7
- Passo 4. Descarregue e execute o programa LEDsSwitches no RVfpgaNexys

## D. Executar e depurar o programa (AL\_Operations) na placa FPGA board

**PRÁTICA:** Siga as instruções da Secção 6.B do Guia de Iniciação (“*AL\_Operations program*”).

## E. Simular o programa AL\_Operations no Whisper

**PRÁTICA:** Siga as instruções da Secção 8 do Guia de Iniciação (“*Simulation in Whisper*”). Salte o passo 5, que se refere ao exemplo HelloWorld que irá testar mais tarde.

## F: Simular o programa AL\_Operations no Verilator

**PRÁTICA:** Siga as instruções da Secção 7 do Guia de Iniciação (“*Simulation in Verilator*”).

### Passo 1. GERAR O BINÁRIO DA SIMULAÇÃO, Vrvfpgasim:

- ESTE PASSO NÃO É NECESSÁRIO NO SEMINÁRIO!
- Os binários são fornecidos em:  
`[RVfpgaPath]/RVfpga/verilatorSIM/OriginalBinaries`

### Passo 2. GERAR O TRACE DE SIMULAÇÃO A PARTIR DO PLATFORMIO, UTILIZANDO O Vrvfpgasim

**Passo 3. ANALISAR O TRAÇO DA SIMULAÇÃO NO GTKWAVE:** Para visualizar os sinais no GTKWave, não siga os passos 10 e 11, mas use o ficheiro *test.tcl* como explicado no passo 12.

## G. Executar o Programa Hello World na Placa

**PRÁTICA:** Siga as instruções da Secção 6.F do Guia de Iniciação (*"HelloWorld\_C-Lang program"*).

Note que no Linux terá de configurar o sistema como explicado no início da secção acima mencionada do GSG.

## H. Simular o Programa HelloWorld no Whisper

**PRÁTICA:** Siga as instruções da Secção 8. Neste caso, tem de substituir nas instruções "AL\_Operations" por "HelloWorld" e ter em consideração as explicações do Passo 5.

## PARTE 1. Labs RVfpga – Parte 1 – Labs 1 a 10

- As instruções e os programas para os laboratórios RVfpga estão disponíveis no seguinte directório, dentro de um subdirectório com o nome do Lab (i.e., “Lab0”, “Lab1”, etc.):

`[RVfpgaPath]/RVfpga/Labs/`

- As soluções utilizadas nos labs RVfpga estão disponíveis em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments`

- Os bitstreams disponíveis em:

Original:

`[RVfpgaPath]/RVfpga/src/rvfpganexys.bit`

Modificado nos Labs 6-10 e Lab 18:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem`

- Os binários do Verilator estão disponíveis em:

`[RVfpgaPath]/RVfpga/verilatorSIM/OriginalBinaries`

### 1. Lab 1 – Programação em C

**DEMONSTRAÇÃO:** Exercício 4. Projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab01/4bitAdd`

**DEMONSTRAÇÃO:** Exercício 9. Projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab01/BubbleSort`

### 2. Lab 2 – Programação em Assembly

**PRÁTICA:** Criar um projecto PlatformIO a partir do zero. Instruções na Secção 3 do documento para o Lab 2.

### 3. Lab 3 – Chamadas a funções

**DEMONSTRAÇÃO:** Exercício 6.

- Este programa utiliza funções para modularizar o programa e bibliotecas C standard (como `srand()` para gerar números aleatórios.).
- Noutros exercícios, também são utilizadas as funções do PSP/BSP da WD (tais como `printfNexys()`).
- Projecto PlatformIO disponível em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab03/SimonSays`

## 4. Lab 4 – Image Processing – C and Assembly

**DEMONSTRAÇÃO:** **Transformação de imagens:** Apresenta a imagem a cores original e a versão transformada da imagem a cinzento. Projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/Lab04/ImageTransformation`

## 5. Lab 5 – Projecto Vivado

**DEMONSTRAÇÃO:** **Secção 2:** Crie um projeto Vivado e gere o bitstream para RVfpgaNexys. Deixe o Vivado a trabalhar em segundo plano e, uma vez terminado, teste o bitstream com o programa LedsSwitches.

## 6. Lab 6 – Introdução às E/S – GPIO

**DEMONSTRAÇÃO:** Reveja o exemplo LedsSwitches do GSG e o Exercício 4 do Lab 1.

**PRÁTICA:** Exercício 5:

- Para executar este programa é necessário utilizar o bitstream modificado com suporte para os cinco botões da placa (conforme orientado nos Exercícios 3 e 4), que é fornecido em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem/RVfpgaSystem_Labs6-10/src/rvfganexys.bit`

- Use o projecto PlatformIO disponível em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab06/Exercise5`

## 7. Lab 7 – Mostradores de 7-Segmentos

**PRÁTICA:** Exercício 1:

- Para executar este programa, é necessário utilizar novamente o bitstream original, que é fornecido em:

`[RVfpgaPath]/RVfpga/src/rvfganexys.bit`

- Use o projecto PlatformIO disponível em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab07/Exercise1`

## 8. Lab 8 – Temporizadores

**DEMONSTRAÇÃO:** Exercício 1.

- Para executar este programa, é necessário utilizar o bitstream original, que é fornecido em:

`[RVfpgaPath]/RVfpga/src/rvfpganexys.bit`

- Projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab08/Exercise1`

### DEMONSTRAÇÃO: Exercício 3.

- Para executar este programa, é necessário utilizar o bitstream modificado com suporte para PWM, que é fornecido em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem/RVfpgaSystem_Labs6-10/src/rvfpganexys.bit`

- Use o projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab08/Exercise3`

## 9. Lab 9 – E/S com Interrupções

Nesta demonstração e exercício prático, utilize novamente o bitstream original, que é fornecido em:

`[RVfpgaPath]/RVfpga/src/rvfpganexys.bit`

### DEMONSTRAÇÃO: Exemplo from Secção 5.

- Primeiro, mostre a execução do programa **LED-Switch\_C-Lang**. Use o projecto PlatformIO provided at:

`[RVfpgaPath]/RVfpga/Labs/Lab09/LED-Switch_C-Lang`

- Depois, mostre a execução do programa **LED-Switch\_7SegDispl\_C-Lang**. Use o projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/Lab09/LED-Switch_7SegDispl_C-Lang`

- Finalmente, mostre a execução do programa **LED-Switch\_7SegDispl\_Interrupts\_C-Lang**. Use o projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/Lab09/LED-Switch_7SegDispl_Interrupts_C-Lang`

**PRÁTICA: Exercício 1.** Teste este exercício, que utiliza interrupções tanto para os interruptores como para o temporizador. Use o projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab09/Exercise1`

## 10. Lab 10 – Barramentos série

### DEMONSTRAÇÃO: Exercício 1.

- Para executar este programa, é necessário utilizar o bitstream original, que é fornecido em:

`[RVfpgaPath]/RVfpga/src/rvfpganexys.bit`

- Projecto PlatformIO disponível em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab10/Accelerometer_7SegDisp`

**DEMONSTRAÇÃO: Exercício 3.**

- Para executar este programa, é necessário utilizar o bitstream original, que é fornecido em:

`[RVfpgaPath]/RVfpga/src/rvfganexys.bit`

- Use o projecto PlatformIO disponível em:


`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab10/Scanf_Printf_7SegDispl`

## PARTE 2. Labs RVfpga – Parte 2 – Labs 11 a 20

### 11. Lab 11 – Configuração, Organização e Monitorização do Desempenho do SweRV EH1

#### PRÁTICA: Tarefa da Secção 2.D:

**TAREFA:** Replicar a simulação da Figura 11 (slide 146) e da Figura 12 (slide 148) no seu próprio computador, seguindo os seguintes passos (conforme descrito em pormenor na Secção 7 do GSG):

- Se necessário, gere o binário de simulação (Vrvfpgasim). Este passo não é necessário no Seminário!!
- No PlatformIO, abra o projecto fornecido em:  
*[RVfpgaPath]/RVfpga/Labs/Lab11/ExampleProgram*.
- Estabeleça o caminho correcto para o binário de simulação RVfpga (Vrvfpgasim) no ficheiro platformio.ini. Lembre-se que os binários são fornecidos em:  
*[RVfpgaPath]/RVfpga/verilatorSIM/OriginalBinaries*
- Gerar o *trace* da simulação com o Verilator (Generate Trace).
- Abra o *trace* usando o GTKWave.
- Use os ficheiros *test\_1.tcl* e *test\_2.tcl* (fornecidos em *[RVfpgaPath]/RVfpga/Labs/Lab11/ExampleProgram*) para abrir os mesmos sinais que os mostrados na Figura 11 e na Figura 12. Para isso, no GTKWave, clique em *File* → *Read Tcl Script File* e seleccione o ficheiro *test\_1.tcl* ou *test\_2.tcl*.
- Clique em *Zoom In* () várias vezes e passar para 48600ps (ou qualquer outra iteração do ciclo, excepto a primeira).

#### PRÁTICA: Tarefa da Secção 3.B:

**TAREFA:** Executar o programa da Figura 13 (slide 151), fornecido em *[RVfpgaPath]/RVfpga/Labs/Lab11/HwCounters\_Example*, na placa Nexys A7 como explicado no GSG. Deve obter os resultados apresentados na Figura 14 para os quatro acontecimentos medidos. Explique e justifique os resultados. Meça outros eventos nos contadores de hardware para o programa da Figura 13.

## 12. Lab 12 – Instruções Aritméticas / Lógicas: a Instrução add

#### DEMONSTRAÇÃO: Exercício 7:

7) (O exercício seguinte baseia-se no exercício 4.4 do [PaHe] e o exercício 1 do capítulo 7 do manual de S. Harris e D. Harris, "Digital Design and Computer Architecture: RISC-V Edition" [DDCARV].)

Quando os chips de silício são fabricados, os defeitos nos materiais (por exemplo, silício) e os erros de fabrico podem resultar em circuitos defeituosos. Um defeito muito comum é um fio de sinal ficar "partido" e registar sempre um 0 lógico, o que é frequentemente designado por defeito "*stuck-at-0*" (preso em 0). Determine o efeito de cada um dos bits de controlo incluídos no sinal *i0\_ap* (um sinal do tipo *alu\_pkt\_t*) ficar preso em 0.

Não efectue a simulação, mostre apenas a solução, que é fornecida em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab12/Lab12_SolutionsTasksAndExercises`

### 13. Lab 13 – Instruções de memória: Instruções `lw` e `sw`

Não há demonstração nem acção prática.

### 14. Lab 14 – Conflitos Estruturais

**PRÁTICA:** Tarefa da Secção 2.A.

**TAREFA:** Remova as instruções `nop` incluídas no ciclo da Figura 1 (apresentada no slide 182) e meça os diferentes eventos (ciclos, instruções/multiplicações cometidas, etc.) usando os Contadores de Desempenho disponíveis no SweRV EH1, como explicado no Laboratório 11. O número de ciclos é o esperado depois de analisar a simulação da Figura 2 (apresentada no slide 183)? Justifique a sua resposta. Agora reordene o código dentro do loop tentando atingir o rendimento ideal. Justifique os resultados obtidos no código original e no código reordenado.

- O programa é fornecido em:  
`[RVfpgaPath]/RVfpga/Labs/Lab14/MUL_Instruction`

### 15. Lab 15 – Conflitos de Dados

**DEMONSTRAÇÃO:** Tarefa da Secção 2.A.

**TAREFA:** Remova as instruções `nop` no exemplo da Figura 2 (apresentado no slide 195). Desenhe uma figura semelhante à Figura 3 (apresentada no slide 196) para duas iterações consecutivas do ciclo, depois analise e confirme que a figura está correcta comparando-a com uma simulação do Verilator e, finalmente, calcule o IPC utilizando os contadores de desempenho enquanto executa o programa na placa.

- Não realize a simulação, apenas mostre a solução que é fornecida em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab15/Lab15_SolutionsTasksAndExercises`

**DEMONSTRAÇÃO:** Tarefa da Secção 3.

**TAREFA:** Desactivar a ALU secundária como explicado no Lab 11 e analise o exemplo da Figura 11 (slide 203) tanto com a simulação no Verilator como com a execução na placa.

- Não realize a simulação, apenas mostre a solução para esta tarefa e a anterior (onde a ALU secundária está activada), que são fornecidas em:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab15/Lab15_SolutionsTasksAndExercises`

## 16. Lab 16 – Conflitos de Controlo & Instruções de Salto

### **PRÁTICA:** Tarefas da Secção 3.

**TAREFA:** No exemplo da Figura 2 (slide 212), retire todas as instruções `nop` e analise a simulação. De seguida, calcule o IPC com os Contadores de Desempenho, executando o programa na placa.

Active o preditor de saltos utilizado no SweRV EH1 (comentando as duas instruções iniciais da Figura 2) e analise a simulação e a execução na placa.

Compare as duas experiências e explique os resultados.

- O programa é fornecido em:  
`[RVfpgaPath]/RVfpga/Labs/Lab16/BEQ_Instruction`
- Não efectue as simulações, mostre apenas as formas de onda, mas execute-as na placa para medir o IPC:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab16/Lab16_SolutionsTasksAndExercises`

## 17. Lab 17 – Superscalar Execution

### **DEMONSTRAÇÕES:** Exercícios 3, 5 e 6.

- Apresente as soluções, que são fornecidas em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/ProgramsAndDocuments/Lab17/Lab17_SolutionsTasksAndExercises`

## 18. Lab 18 – Adding New Features

### **PRÁTICA:** Exercícios 1, 3 e 4.

#### **Exercício 1:**

- Leia a primeira parte do enunciado do exercício para compreender o processo de inclusão de uma nova instrução no núcleo do SweRV EH1.
- Note que não precisa de seguir o processo explicado, uma vez que o bitstream do processador, incluindo as novas instruções, é fornecido nas soluções.

#### **Exercício 3:**

- Leia o enunciado do exercício. Não efectue as simulações.
- Em seguida, analise e teste na placa o programa de:

`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem/RVfpgaSystem_Lab18/FloatingPoint/DotProduct_Comparision`

- Para o efeito, pode utilizar o mesmo bitstream que está disponível em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem/RVfpgaSystem_Lab18/FloatingPoint/src`
- Compare os resultados do produto escalar calculado em software e em hardware: **`sum_swemul`** vs. **`sum_hwimpl`**.
- Compare o número de ciclos de execução em software e em hardware.

#### **Exercício 4:**

- Leia o enunciado do exercício.
- Analise e teste na placa o programa de:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem/RVfpgaSystem_Lab18/FloatingPoint/BisectionAlgorithm`
- Para o efeito, pode utilizar o mesmo bitstream do Exercício 3, que está disponível em:  
`[RVfpgaPath]/RVfpga/Labs/RVfpgaLabsSolutions/Modified_RVfpgaSystem/RVfpgaSystem_Lab18/FloatingPoint/src`
- Compare os resultados do produto escalar calculado em software e em hardware: **`zero_swemul`** vs. **`zero_hwemul`**.
- Compare o número de ciclos de execução em software e em hardware.

## **19. Lab 19 – Cache de instruções**

Não há demonstração nem acção prática.

## **20. Lab 20 – ICCM, DCCM & Avaliação de Desempenho (Benchmarking)**

### **PRÁTICA: Secção 3**

- **Secção 3.A:** Sem optimizações do compilador ou DCCM/ICCM
- **Secção 3.B:** Utilização da DCCM
- **Secção 3.C:** Utilização da DCCM e com optimizações do compilador