CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES II - Turma 01

PRÁTICA 4 - PROTOCOLO DE COERÊNCIA SNOOPING

Natan Rodrigues Rocha / 20203013242
Gabriel Andrade Quezada / 20183021240

Belo Horizonte
10 de Setembro de 2024

PARTE 1

INTRODUÇÃO

A primeira etapa deste trabalho consistiu na simulação de todas as transições de estado das duas máquinas de estado envolvidas no protocolo Snooping: emissora e receptora. Para isso, foram desenvolvidos módulos específicos para cada uma dessas máquinas, que gerenciam e processam as mensagens trocadas no sistema.

O módulo responsável pelo Emissor recebe sinais de entrada, como *read miss*, *read hit*, *write hit* e *write miss*, que ativam diferentes transições na sua máquina de estados. A resposta gerada por essa máquina é então enviada ao módulo Receptor, que processa mensagens de *write miss*, *read miss* e *invalidate*, coordenando as ações necessárias para garantir a consistência da memória. Além disso, o Receptor exibe todas as mensagens de *writeback* e *abort*, essenciais para a comunicação adequada entre as máquinas.

SIMULAÇÃO DAS TRANSIÇÕES DE ESTADO NO PROTOCOLO SNOOPING

Os resultados dessas transições são apresentados tanto em forma de ondas, onde as transições da emissora aparecem em vermelho e as da receptora em dourado, quanto em forma de tabela. Abaixo, são listados os principais mapeamentos utilizados:

Estados:

- INVÁLIDO = 0
- COMPARTILHADO = 1
- MODIFICADO = 2

Mensagens:

- ERRO DE LEITURA = 1
- ERRO DE ESCRITA = 2
- INVALIDAR = 3

Ações:

- WRITEBACK = 1
- ABORT = 2
- WRITEBACK E ABORT = 3

Este modelo permite uma compreensão clara e detalhada das transições de estado e das trocas de mensagens, facilitando a análise do comportamento do protocolo e a verificação da consistência da memória em sistemas multiprocessados. A utilização e

criação de testbenches foi essencial para nos ajudar a simular com êxito, sendo uma boa ferramenta aprendida em LAOC1, ainda que não seja sempre a ideal.

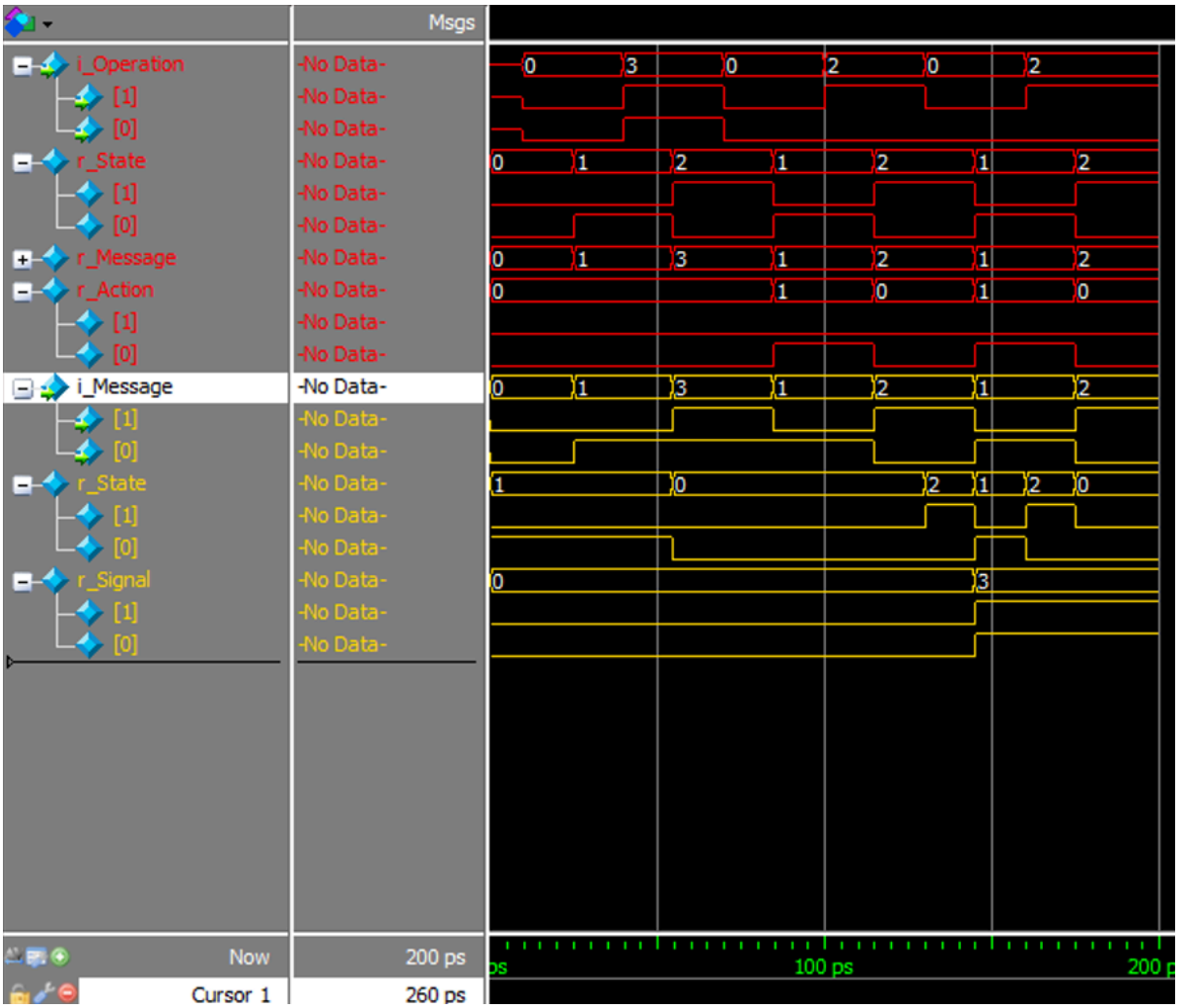

Figura 01 - Formas de onda

Máquina	Estado	Mensagem	Ação	Sinal(Receptora)	Operação
Emissora	0	0	0	-	**RM**: Move emissora para SHARED e receptora para SHARED
Receptora	1	-	-	0	

Máquina	Estado	Mensagem	Ação	Sinal(Receptora)	Operação
Emissora	1	1	0	-	**WH**: Move emissora para MODIFIED e receptora para INVALID
Receptora	1	-	-	0	

Máquina	Estado	Mensagem	Ação	Sinal(Receptora)	Operação
Emissora	2	3	0	-	**RM**: Move emissora para SHARED, ativa WRITEBACK, e receptora para INVALID
Receptora	0	-	-	0	

Máquina	Estado	Mensagem	Ação	Sinal(Receptora)	Operação
Emissora	1	1	1	-	**WM**: Move emissora para MODIFIED e receptora para INVALID
Receptora	0	-	-	0	

Máquina	Estado	Mensagem	Ação	Sinal(Receptora)	Operação
Emissora	2	2	0	-	**RM**: Move emissora para SHARED, faz WRITEBACK e receptora para SHARED, ativando ABORT

Receptora	0	-	-	3	

Máquina	Estado	Mensagem	Ação	Sinal(Receptora)	Operação
Emissora	1	1	1	-	**WM**: Move emissora para MODIFIED e receptora para INVALID, ativando ABORT
Receptora	1	-	-	3	

Legenda:

- **RM** = *Read Miss*
- **WH** = *Write Hit*
- **WM** = *Write Miss*
- **SHARED** = Estado Compartilhado
- **MODIFIED** = Estado Modificado
- **INVALID** = Estado Inválido
- **WRITEBACK** = Escrita de volta
- **ABORT** = Abortar

Essa tabela ilustra as principais operações do protocolo Snooping, mostrando as transições dos estados da máquina emissora e receptora de acordo com os sinais de entrada e as mensagens trocadas.

CONCLUSÃO

A simulação realizada no ModelSim foi bem-sucedida, demonstrando o comportamento esperado das máquinas de estado no protocolo Snooping. No entanto, apesar dos resultados promissores no ambiente de simulação, visto que são apenas mudanças de estados, a implementação e os testes realizados na FPGA não obtiveram o mesmo êxito. Foram conduzidos diversas alterações de código com criação de novos módulos para teste na placa, mas como não foi possível replicar os resultados observados na simulação por falta de tempo e erros inesperados na sintaxe, essa parte foi deixada de lado. Esse contraste indica a necessidade de revisões adicionais na integração com a FPGA para garantir o correto funcionamento do sistema em hardware.

PARTE 2

INTRODUÇÃO

O projeto Snooping é uma técnica essencial para garantir a consistência dos dados entre as caches de diferentes processadores e a memória principal. O princípio básico é que os processadores compartilhem um barramento comum, permitindo a comunicação de estados e a troca de informações. Sempre que um processador altera um dado, todos os outros que possuem uma cópia da mesma tag na cache são notificados e precisam realizar uma ação correspondente, como invalidar ou atualizar seus dados, para manter a coerência.

A solução desenvolvida segue o modelo tradicional estudado em sala de aula, mas com algumas modificações no barramento de comunicação, adaptadas para melhorar a eficiência e o desempenho do sistema. As mudanças implementadas serão descritas em detalhes no tópico seguinte.

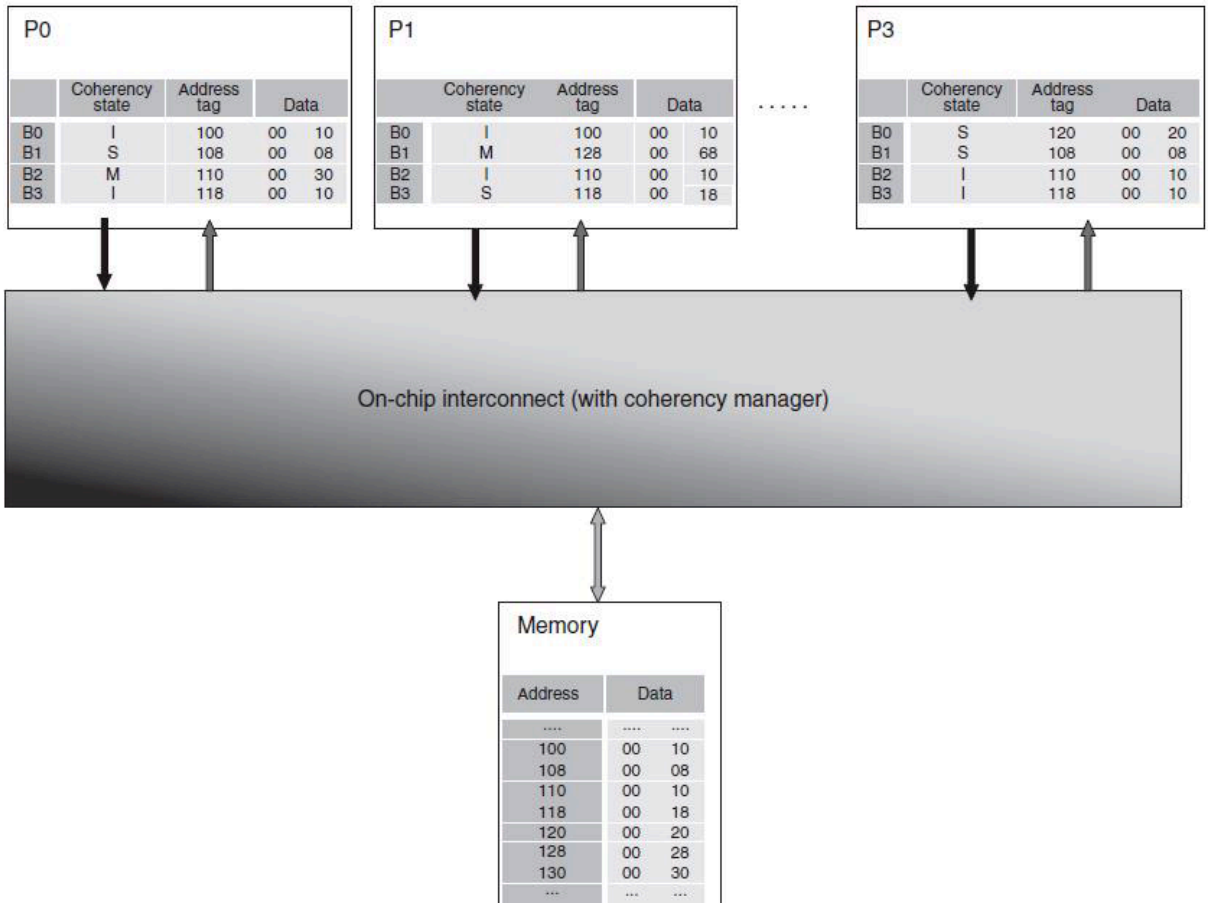

Figura 01 - Modelo do projeto

PROJETO

O módulo principal do projeto Snooping é responsável por instanciar os quatro processadores, a memória compartilhada e o barramento de comunicação. Ele atua como a base do sistema, onde a coerência de cache entre os processadores é garantida.

Estrutura dos Módulos

1. **Bus (Barramento de Comunicação)** O módulo do barramento (*Bus*) é responsável por receber as saídas dos processadores e da memória, encaminhando essas informações para os demais componentes. Sua implementação inclui um único bloco *always*, que verifica qual entrada não está nula e a propaga para os outros componentes através do barramento. A seguir, é mostrado o mapeamento dos bits no barramento:
    - **bit 23**: Indica se o emissor deve usar o bloco modificado.
    - **bits 22-21**: Indicam a mensagem.
    - **bits 20-18**: Indicam a tag envolvida.
    - **bits 17-11**: Representam o dado.
    - **bit 10**: Indica se é preciso realizar *writeback*.
    - **bits 9-7**: Indicam a tag a ser feito o *writeback*.
    - **bits 6-0**: Representam o dado a ser feito *writeback*.
2. **Memory (Memória Compartilhada)** A memória é compartilhada por todos os processadores, com um registrador de 10x4 bits para armazenar os dados. Uma modificação significativa no projeto, em relação ao modelo tradicional do Snooping, está relacionada ao gerenciamento simultâneo de escritas na memória e à comunicação entre processadores. Originalmente, ocorria um problema ao tentar realizar ambas as operações ao mesmo tempo. Para resolver isso, o barramento foi duplicado, permitindo o envio simultâneo das informações para a memória e os processadores.
3. **Cache (Processadores)** Cada processador possui sua própria cache, identificada de forma única no sistema. A cache de cada processador tem uma estrutura de 12x3 bits, conforme descrito abaixo:
    - **bits 11-10**: Indicam o estado atual da cache (inválido, compartilhado ou modificado).
    - **bits 9-7**: Indicam a tag correspondente ao dado.
    - **bits 6-0**: Representam o dado armazenado.
4. A lógica da cache é dividida em dois blocos *always* distintos:
    - O primeiro bloco representa a máquina de estados do emissor, sensível ao *clock* para que as operações sejam executadas sequencialmente.
    - O segundo bloco é a máquina de estados do receptor, sensível ao barramento de dados, que processa as mensagens recebidas dos outros processadores ou da memória.

Processamento das Instruções

As instruções chegam como entrada na cache e são decompostas em seus respectivos bits, conforme o mapeamento descrito. A máquina de estados do emissor foi dividida em três etapas para lidar com o tempo necessário para que os dados da memória fiquem prontos. As etapas 2 e 3 servem para aguardar a finalização do processo e, por fim, atualizar os valores nas caches dos processadores.

Considerações de Síntese

Vale destacar que, embora o código desenvolvido não seja sintetizável, devido à afinidade dos registradores com blocos *always* (que não é respeitada nesta implementação), isso não impede o sucesso da simulação, que é o objetivo principal deste trabalho.

SIMULAÇÃO

O código de testes utilizado corresponde ao apresentado na aula teórica. As instruções são as seguintes:

```
13'b0001000000000; // P0: READ 120
13'b1001001010000; // P0: WRITE 120 <- 80
13'b1111001010000; // P3: WRITE 120 <- 80
13'b0010100000000; // P1: READ 110
13'b1110100011110; // P3: WRITE 110 <- 30
13'b0110100000000; // P3: READ 110
13'b1000010110000; // P0: WRITE 108 <- 48
13'b1111101001110; // P3: WRITE 130 <- 78
```

A configuração de bits para as instruções é a seguinte:

- **bit 12**: 0 para leitura e 1 para escrita.
- **bits 11-10**: identificam o processador.
- **bits 9-7**: tag.
- **bits 6-0**: valor.

Os estados iniciais da cache estão armazenados em arquivos .mem, e o Modelsim requer o caminho absoluto desses arquivos, devendo ser ajustado caso necessário. Como a visualização das ondas na simulação é complexa, os displays serão usados para facilitar a interpretação. No entanto, as formas de onda também estão disponíveis nos arquivos gerados.

Os estados da cache seguem o mapeamento:

- INVALID = 0
- SHARED = 1
- MODIFIED = 2

Execução dos comandos:

P0: READ 120

Alteração:

- P0.B0(S,120,0020)

C0
CONTENTS:
B0: S 1 T 4 V
20
B1: S 1 T 1 V 8
B2: S 2 T 2 V 30
B3: S 0 T 3 V 10

C1
CONTENTS:
B0: S 0 T 0 V
10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 10
B3: S 1 T 3 V 18

C3
CONTENTS:
B0: S 1 T 4 V
20
B1: S 1 T 1 V 8
B2: S 0 T 2 V 10
B3: S 0 T 3 V 10

MEM

CONTENTS:

T 1 V 8
T 2 V 10
T 4 V 20
T 6 V 30

P0: WRITE 120 <- 80
Alteração:

- P0.BO(M,120,0080)
- P3.BO(1,120,0020)

C0 CONTENTS:
B0: S 2 T 4 V 80
B1: S 1 T 1 V 8
B2: S 2 T 2 V 30
B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 10
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 0 T 4 V 20
B1: S 1 T 1 V 8
B2: S 0 T 2 V 10
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 10
T 4 V 20
T 6 V 30

P3: WRITE 120 <- 80
Alteração:

- P3.BO(M,120,0080)
- P0.BO(I,120,0080)
- M(120,0080)

C0 CONTENTS:
B0: S 0 T 4 V 80
B1: S 1 T 1 V 8
B2: S 2 T 2 V 30

B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 10
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 2 T 4 V 80
B1: S 1 T 1 V 8
B2: S 0 T 2 V 10
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 10
T 4 V 80
T 6 V 30

P1: READ 110
Alteração:

- P1.B2(S,110,0030)
- P0.B2(S,110,0030)
- M(110,0030)

C0 CONTENTS:
B0: S 0 T 4 V 80
B1: S 1 T 1 V 8
B2: S 1 T 2 V 30
B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 1 T 2 V 30
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 2 T 4 V 80
B1: S 1 T 1 V 8
B2: S 0 T 2 V 10
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 30
T 4 V 80
T 6 V 30

P3: WRITE 110 <- 30

Alteração:

- P1.B2(I,110,0030)
- P0.B2(I,110,0030)
- P3.B2(M,110,0030)

C0 CONTENTS:
B0: S 0 T 4 V 80
B1: S 1 T 1 V 8
B2: S 0 T 2 V 30
B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 30
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 2 T 4 V 80
B1: S 1 T 1 V 8
B2: S 2 T 2 V 30
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 30
T 4 V 80
T 6 V 30

P3: READ 110
Alteração (apenas lê)

C0 CONTENTS:
B0: S 0 T 4 V 80
B1: S 1 T 1 V 8
B2: S 0 T 2 V 30
B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 30
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 2 T 4 V 80
B1: S 1 T 1 V 8
B2: S 2 T 2 V 30
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 30
T 4 V 80
T 6 V 30

P0: WRITE 108 <- 48
Alteração:

- P0.B1(M,108,0048)
- P3.B1(I,108,0008)

C0 CONTENTS:
B0: S 0 T 4 V 80
B1: S 2 T 1 V 48
B2: S 0 T 2 V 30
B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 30
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 2 T 4 V 80
B1: S 0 T 1 V 8
B2: S 2 T 2 V 30
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 30
T 4 V 80
T 6 V 30

P0: WRITE 130 <- 78
Alteração:

- P0.B2(M,130,0078)

C0 CONTENTS:
B0: S 0 T 4 V 80
B1: S 2 T 1 V 48
B2: S 0 T 2 V 30
B3: S 0 T 3 V 10

C1 CONTENTS:
B0: S 0 T 0 V 10
B1: S 2 T 5 V 68
B2: S 0 T 2 V 30
B3: S 1 T 3 V 18

C3 CONTENTS:
B0: S 2 T 4 V 80
B1: S 0 T 1 V 8
B2: S 2 T 6 V 78
B3: S 0 T 3 V 10

MEM
CONTENTS:
T 1 V 8
T 2 V 30
T 4 V 80
T 6 V 30

CONCLUSÃO

DIFICULDADES ENCONTRADAS

Durante o desenvolvimento do projeto, enfrentamos algumas dificuldades significativas que impactaram o andamento e a conclusão da prática:

1. **Controle do Barramento de Dados:** A principal dificuldade foi o controle do barramento de dados, essencial para garantir que as informações fossem enviadas no momento correto para os processadores apropriados. Algo que na teoria é tranquilo, mas, na prática, se torna um problema. Esse problema exigiu várias pequenas alterações no sistema para assegurar que o barramento funcionasse de maneira eficaz. Ajustes foram necessários para sincronizar corretamente o envio e o recebimento dos dados entre os diferentes processadores.
2. **Envio de Blocos Modificados:** Outro desafio enfrentado foi o envio de blocos modificados de um processador de volta para o receptor. Inicialmente, houve problemas na integração e na correta identificação dos blocos modificados, o que gerou inconsistências e falhas na comunicação. Após a identificação do problema, foram realizadas correções que permitiram que o envio dos blocos modificados fosse realizado de maneira adequada.
3. **Prazo e Apresentação:** A prática enfrentou diversos erros durante seu desenvolvimento, e devido à complexidade dos problemas encontrados, aliado a um final de semestre corrido com todas as outras matérias, o projeto não foi concluído a tempo para a apresentação marcada. Como resultado, a apresentação para a professora foi impossibilitada. Apesar das dificuldades, após muita pesquisa e muito estudo da teoria, correções necessárias foram feitas, e o projeto foi finalizado após o prazo estabelecido.

SUGESTÕES

A prática de snooping foi desenvolvida não da maneira ideal, mas segundo os testes, funcionando e, portanto, há pouco a ser alterado. No entanto, talvez seja interessante considerar a adição de algumas dicas para os módulos ou exemplos, a fim de melhorar a experiência e a eficácia dessa prática, além de facilitar no desenvolvimento da mesma. Acreditamos que essas sugestões poderiam contribuir para otimizar o aprendizado e o desempenho dos alunos.